The Canadian Journal of
Chemical Engineering

Experimental Methods in Chemical Engineering: Process Simulation

Journal:	*The Canadian Journal of Chemical Engineering*
Manuscript ID	CJCE-20-0335.R1
Wiley - Manuscript type:	Mini-Review
Date Submitted by the Author:	19-Jun-2020
Complete List of Authors:	De Tommaso, Jacopo; Montreal Polytechnic, Chemical Engineering Rossi, Francesco; Purdue University, School of Chemical Engineering, Davidson School of Chemical Engineering, Forney Hall of Chemical Engineering Moradi, Nooshin; Shiraz University, School of Chemical and Petroleum Engineering Pirola, Carlo; Universita degli Studi di Milano, Dipartimento di Chimica Patience, Gregory; Montreal Polytechnic, Genie Chimique Galli, Federico; Montreal Polytechnic, Chemical Engineering
Keywords:	Process Simulation, Design, Thermodynamics, Optimization, Cost Estimation

SCHOLARONE™
Manuscripts

- Simulation optimizes, designs, estimates costs and is a teaching tool for engineers
- Simulations reduces experiments and indicates thermodynamic boundaries
- Users are the main errors source: data input, model selected, wrong specifications
- Proc Sim clusters: design, optimization, CO_2 capture, biomass, and gasification

For Peer Review

Received ; | Revised: 00 Month 0000 | Accepted: 00 Month 0000

DOI: xxx/xxxx

MINI-REVIEW

Experimental Methods in Chemical Engineering: Process Simulation

Jacopo De Tommaso[1] | Francesco Rossi[2] | Nooshin Moradi[3] | Carlo Pirola[4] | Gregory S. Patience[1] | Federico Galli*[1]

[1]Chemical Engineering, Polytechnique Montréal, C.P. 6079, Succ. "CV", Montréal, H3C 3A7 Québec, Canada
[2]School of Chemical Engineering, Pordue University, Davidson School of Chemical Engineering, Forney Hall of Chemical Engineering, 480 Stadium Mall drive, West Lafayette, 47907-2100 IN, USA
[3]School of Chemical and Petroleum Engineering, Shiraz University, 71348-51154, Shiraz Iran
[4]Dipartimento di Chimica, Universitá degli Studi di Milano, via Golgi 19, 20133, Milano, Italy

Correspondence
*Corresponding author: Federico Galli, Polytechnique Montréal. Email: federico.galli@polymtl.ca

Summary

Process simulation software designs equipment, simulates operations, optimizes a plant's configuration (heat exchangers network, for example), estimates operating and capital expenses, and serves as educational tools. However, mastering the theoretical background minimizes common mistakes such as applying an incorrect thermodynamic method, selecting improper algorithms in the case of tear systems, and set irrational system specifications. Engineers and researchers will exploit this tool more often in the future as constant advancements in simulation science as well as new models are released continually. Process simulators ease the building of digital twins and thus will facilitate implementation of the industry 4.0 guidelines. We highlight the mathematical and technical features of process simulators, as well as the capabilities and the fields of applications. A bibliometric map of keywords from articles citing Aspen+, Aspen plus, Hysys, and Pro/II indexed by *Web of Science* between 2017 and 2020 identified the main research clusters such as: Design; Optimization; Energy or Exergy; Biomass; H_2 and CO_2 capture; Thermodynamics; separations and Techno-Economic analysis.

KEYWORDS:
Process simulation, Design, Thermodynamics, Optimization, Cost estimation

1 | INTRODUCTION

Process simulators solve mathematical equations not only to characterize reactors, distillation columns, heat exchangers, pumps, compressors and other unit operations but also for process design and optimization. Furthermore, they offer engineers a tool to estimate plant costs—both operating expenses (OPEX) and capital expense (CAPEX). Users input operating pressure, temperature, flow rates, composition, and thermodynamic packages and the software computes mass and energy balances for each stream and module. Examples of the first software date back to 1960s with PACER developed by McMaster University for educational purposes. The University of Houston developed in collaboration with industry CHESS in 1968. It simulated the phase equilibria of more than 70 hydrocarbons.[1] PROCESS, by Simulation science, was distributed in 1966 and simulated distillation columns. It became then PRO/II, first distributed by Scheider-Electrics and then by AVEVA. Fortran expanded the capabilities of these programs in the 70s. In the late 70s, MIT developed Advanced System for Process ENgineering (ASPEN) that AspenTech™ commercializes in the 80s. Concurrently Profs. Bishnoi and Svercek at the University of Calgary developed HYSYS. In the late 1980s and early 1990s the PC-based simulators were developed.[2] In 2002 AspenTech™ acquired Hyprotech; however, the Federal Trade Commission required AspenTech to divest the HYSYS because they deemed it contravened anti-trust laws. Honeywell

For Peer Review

became the owner of HYSYS and created UniSim. Eventually, AspenTech™ re-acquired HYSYS in 2016 and now 91 simulators are on the market including open source software.

This mini-review on process simulation is part of a series of articles dedicated to experimental methods in chemical engineering.[3] Although practicing engineers and students use these simulators for plant design and optimization, researchers also apply them to calculate thermodynamics, identify equilibrium compositions of reactions, verify the feasibility of separation operations, and thus optimize experimental designs. Here we discuss these features but address the mathematical approaches to solve these large systems of equations, resolution algorithms, and applications. We describe the main sources of error and how to avoid common mistakes that arise at the set up of the flowsheet and include a bibliometric survey that highlights the major applications.

2 | THEORY

Steady-state models of most unit operations are nonlinear systems of either algebraic or differential-algebraic equations (e.g., PFR reactors and pipes). Discretization techniques convert differential -algebraic equations into a system of algebraic equations, so any steady-state process flowsheet is equivalent to a large-scale system of nonlinear equations. Therefore, in this section of the manuscript, we first review the rationale of methods that solve these systems of nonlinear equations, and then explain how simulation packages use them to solve process flowsheets. We discuss both modular and equation-oriented strategies for flowsheet simulation, comment on their advantages and disadvantages, and provide guidelines on when/how to apply each approach.[4, 5]

2.1 | Numerical methods for systems of nonlinear equations

The most general mathematical formulation of a system of nonlinear equations, Eq 1, where every function g_i represents a nonlinear expression of the unknown variables x_i (in a process flowsheet, the functions g_i represent steady-state mass, energy and momentum balances, and discretization thereof or design specifications, whereas the quantities x_i are process variables, e.g. temperatures, pressures, molar/mass fractions and flowrates).

$$\begin{cases} g_1(x_1, x_2, x_3, ..., x_N) = 0 \\ g_2(x_1, x_2, x_3, ..., x_N) = 0 \\ \cdots \\ g_N(x_1, x_2, x_3, ..., x_N) = 0 \end{cases} \Leftrightarrow g(\boldsymbol{x}) = 0 \qquad (1)$$

The objective to solve this nonlinear system ($\boldsymbol{x_s}$), which corresponds to a root of all the functions g_i ($g_i(\boldsymbol{x_s}) = 0$). There exist no analytical method to accomplish this task, so we only estimate an approximate value of $\boldsymbol{x_s}$ through numerical algorithms—Newton type and fixed-point methods. Both these iterative procedures rely on a user-supplied initial value, $\boldsymbol{x_0}$ and iteration $(\boldsymbol{x_n})_{n\geq1}$ until they approach the solution vector $\boldsymbol{x_s}$. These two families of algorithms differ with respect to the procedure they apply to compute $(\boldsymbol{x_n})_{n\geq1}$, their efficiency and robustness, and their application domain.

Starting from the last known iteration, $\boldsymbol{x_n}$, Newton-type methods calculate $\boldsymbol{x_{n+1}}$ with the following procedure:

1. Exact or approximate linearization of the nonlinear systems at $\boldsymbol{x_n}$. This step requires calculation of the Jacobian matrix at $\boldsymbol{x_n}$ or of an estimate thereof (the matrices $\boldsymbol{J}(\boldsymbol{x_n})$ and B_n, respectively);

2. Solution of the linearized system of equations and calculation of the search direction $\boldsymbol{d_n}$, according to the first line of Eq. 2 and Eq. 3;

3. Computation of the optimal step length $\boldsymbol{a_n}$ through a line search strategy, performed along $\boldsymbol{d_n}$;[6]

4. Calculation of $\boldsymbol{x_{n+1}}$ from $\boldsymbol{x_n}$, $\boldsymbol{a_n}$, and $\boldsymbol{d_n}$, according to the second line of Eq. 2 and Eq. 3.

$$\begin{cases} \boldsymbol{J}(\boldsymbol{x_n})\boldsymbol{d_n} = -g(\boldsymbol{x_n}) \\ \boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{a_n}\boldsymbol{d_n} \end{cases} \wedge \boldsymbol{J}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_N} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_N}{\partial x_1} & \frac{\partial g_N}{\partial x_2} & \cdots & \frac{\partial g_N}{\partial x_N} \end{pmatrix} \qquad (2)$$

$$\begin{cases} B_n\boldsymbol{d_n} = -g(\boldsymbol{x_n}) \\ \boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{a_n}\boldsymbol{d_n} \end{cases} \wedge B(n) \approx \boldsymbol{J}(\boldsymbol{x_n}) \qquad (3)$$

These steps are repeated until convergence criteria are met, e.g. the magnitude of the search direction ($\|\boldsymbol{d_n}\|$) becomes sufficiently small, or the sequence $(\boldsymbol{x_n})_{n\geq1}$ diverges (Fig. 1). The sequence $(\boldsymbol{x_n})_{n\geq1}$ follows the same type of pattern (Fig. 1) for systems of any number of equations.

Newton-type methods do not guarantee convergence to a solution of the nonlinear system, starting from any random initial guess as this family of algorithms is only locally convergent. However, their rate of convergence is superlinear. Therefore, they are very effective when a good initial guess is available. Another feature of Newton-type methods is their capability of solving nonlinear systems of *arbitrarily* large scale, provided that adequate computing power is available.

Fixed-point methods require a reformulation of the original system of nonlinear equations, which consists of re-writing

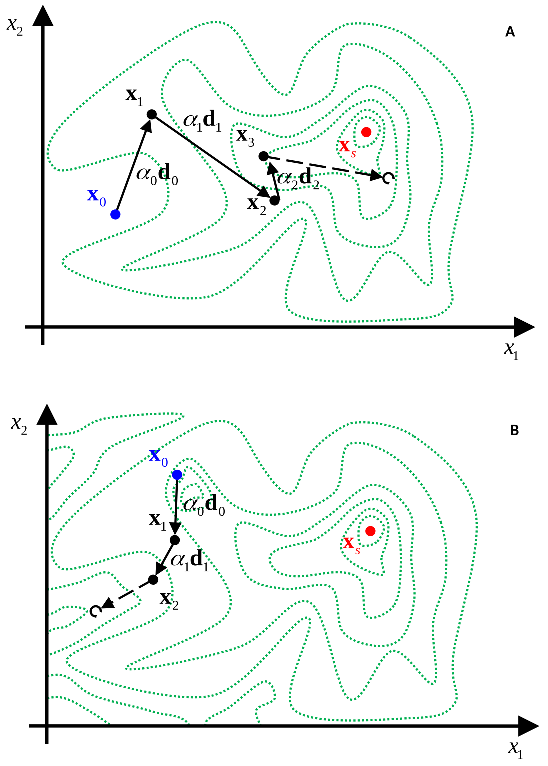

FIGURE 1 Typical sequence of iterations generated by a Newton-type method. (A - convergent sequence; B - divergent sequence). From the initial guess $\boldsymbol{x_0}$, the algorithm calculates the direction vector $\boldsymbol{d_i}$ and the step length $\boldsymbol{a_i}$, moving towards $\boldsymbol{x_S}$ or without converging.

every equation $g_i(\boldsymbol{x}) = 0$ in the same form as Eq. 4 (this reformulation is always possible, although not necessarily unique, and may affect the convergence properties of the algorithm).

$$x_i = f_i(\boldsymbol{x}) \quad \forall i \in [1, N] \qquad (4)$$

After this preliminary step, starting from the last known iteration $\boldsymbol{x_n}$, the next iteration $\boldsymbol{x_{n+1}}$ is calculated through the following procedure:

1. Calculation of the functions f_i at $\boldsymbol{x_n}$;
2. Selection of values for all the relaxation factors $\boldsymbol{a_{i,n}}$;
3. Application of Eq. 5.

$$\boldsymbol{x_{i,n+1}} = \boldsymbol{a_{i,n}} f_i(\boldsymbol{x_n}) + (1 - \boldsymbol{a_{i,n}})\boldsymbol{x_{i,n}} \quad \forall i \in [1, N] \qquad (5)$$

These steps are repeated until convergence criteria are met, e.g. the norm of the difference between two consecutive iterations ($\|\boldsymbol{x_{n+1}} - \boldsymbol{x_n}\|$) becomes sufficiently small, or the sequence $((\boldsymbol{x_n})_{n\geq 1}$ diverges (the direct substitution recursion matches Eq. 5 with all the relaxation factors $\boldsymbol{a_{i,n}} = 1$). The sequence $(\boldsymbol{x_n})_{n\geq 1}$ follows a similar pattern for systems of two or more equations (Fig. 2).

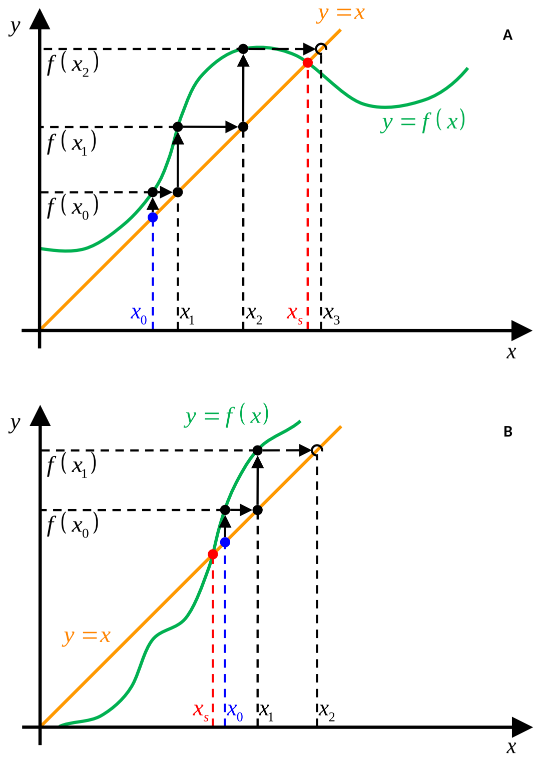

FIGURE 2 Typical sequence of iterations generated by direct substitution (A–convergent sequence; B–divergent sequence). From the initial guess $\boldsymbol{x_0}$, the algorithm calculates $\boldsymbol{x_i}$ and $f(\boldsymbol{x_i})$, moving towards $\boldsymbol{x_S}$ or without converging.

As with Newton-type methods, fixed-point schemes do not guarantee convergence to a solution of the nonlinear system, starting from any random initial guess (these algorithms are only locally convergent), but they are usually less sensitive to the initial value. On the other hand, their rate of convergence is slower than Newton-type methods. Thus, they are particularly suitable for situations in which no good initial guess can be computed. Note that, as opposed to Newton-type algorithms, fixed-point methods can only solve small-scale/medium-scale nonlinear systems, regardless of the amount of computing power available. This limitation considerably restricts their application domain.

2.2 | Computational Strategy

The barebones of a flowsheeting software differs from those of others based on the computational strategies. The main techniques are:

- Sequential Modular Approach (**SM**)
- Equation Oriented Approach (**EO**)
- Simultaneous Modular Approach

The SM solves the process units in sequence, starting from the feed and tear the common streams in case of recycle. Where an inlet stream is given, each block computes its outlet stream. However, in the presence of a recycle, an iteration is required. Very sturdy and reliable, this is the default method of commercial and general software, especially for steady-state calculations. On the other hand, The SM is weak and time-consuming for[7]:

- Highly recycled processes;
- Highly-Heat integrated processes;
- Optimization;
- Simultaneous flowsheet and design specification loops; and,
- Dynamic simulations.

In the EO approach the whole process is solved simultaneously as a system of nonlinear algebraic equation. EO requires a more experienced simulator, is difficult to debug, and works well only when the initial value is close to the solution. Nevertheless, it works better where SM is weak. The Simultaneous Modular Approach is a combination of both SM and EO.[8] The flowsheet is solved in an EO fashion, while the singular unit are solved sequentially.[9] So far, this approach finds application in academia,[10–12] but not in commercial software.

2.3 | Modular solution (SM) strategies for process flowsheets

Any chemical process is a set of unit operations, primarily connected in series as the number of recycles is usually significantly smaller than the total number of connections. Modular solution strategies for process simulation leverage this sequential nature of chemical processes to first simplify and then perform flowsheet calculations. The rationale of these solution strategies is:

1. Given a certain process flow diagram (PFD), we first convert it into a directed graph (the simulation flow diagram (SFD)), whose vertices and edges represent units and streams, respectively (Fig. 3).
2. Then, well-established algorithms, e.g. the connection matrix method[13]or the depth-first search and backtrack method[14], detect all the simple cycles in the flowsheet graph (Fig. 3), which corresponds to identifying all the recycles in the original process flowsheet.
3. Subsequently, we break all the simple cycles by tearing a number of edges in the flowsheet graph (Fig. 4), thus transforming the latter into a directed tree (the flowsheet tree). This graphical operation corresponds to breaking all the recycles in the original process flowsheet, by replacing specific process streams (the tear streams) with pairs of new streams (the artificial streams), so as to generate a simplified process flowsheet, in which all the unit operations can be solved in series, once half the artificial streams have been assigned (the remaining artificial streams become simulation outputs). Note that every recycle stream is literally torn at least once, thus the name *tear stream*.
4. Finally, we exploit the connectivity information, stored in the flowsheet tree, to solve all the unit operations and compute all the process streams in the simplified process flowsheet, under the additional requirement that every pair of artificial streams be identical (these artificial constraints ensure that all the recycles are enforced). This is mathematically equivalent to solving Eq. 6 (the recycle problem), in which the vector $\boldsymbol{x}$ represents the process variables of the artificial streams that must be assigned, and the function $f(\boldsymbol{x})$ denotes the process variables of the artificial streams that are simulation outputs. $f(\boldsymbol{x})$ is an implicit function of $\boldsymbol{x}$, as we can only evaluate it by solving some/all of the unit operations in the simplified process flowsheet).

$$\boldsymbol{x} - f(\boldsymbol{x}) = 0 \tag{6}$$

Eq. 6 contains only a small fraction of the nonlinear equations, which describe the original process flowsheet of interest, provided that the latter contains a small number of recycles (the scale of Eq. 6 is indeed proportional to the number of recycles). Therefore, modular solution strategies are suitable for simulation of process flowsheets of *arbitrary* scale, which contain few recycles. On the other hand, they should never be used to solve process flowsheets with many recycles, as the numerical solution of Eq. 6 becomes impractical.

2.3.1 | Selection of the optimal tear streams

The identification of optimal tear streams is an element of any modular solution strategy for process flowsheets because the number and features of the tear streams influence the size and numerical properties of the recycle problem (Eq. 6). Therefore,

For Peer Review

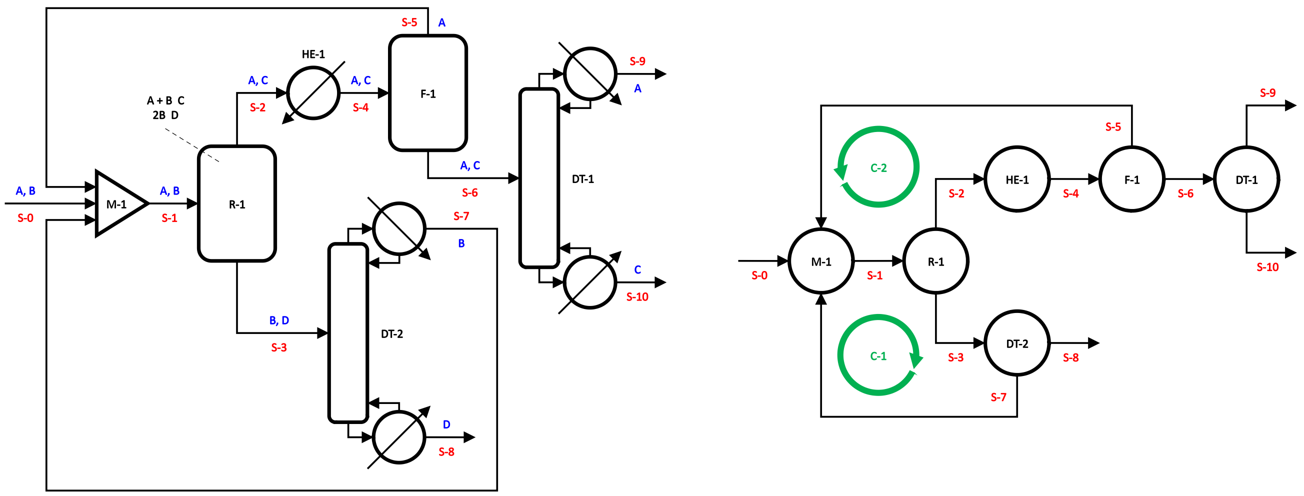

FIGURE 3 Simple process flowsheet complemented with its graph representation (S-N – Nth process stream; C-N – Nth simple cycle or Nth recycle).

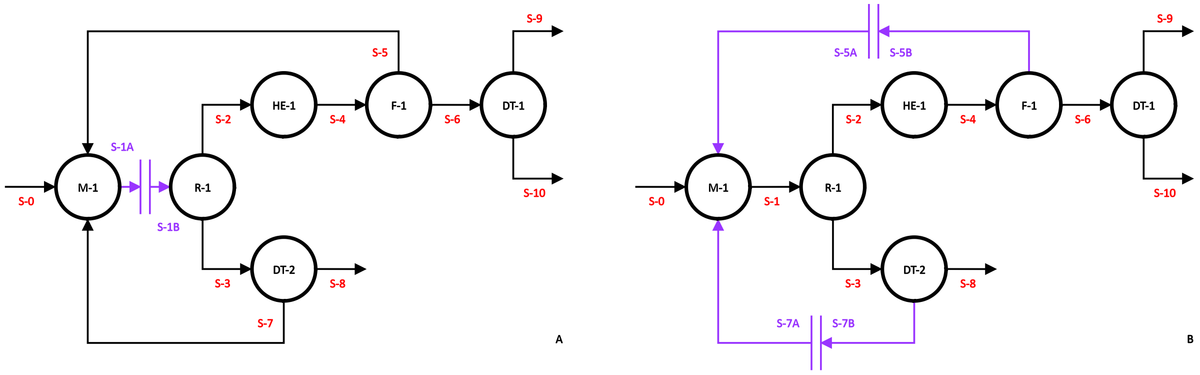

FIGURE 4 Reduction of the flowsheet graph to a directed tree by elimination of all the simple cycles (A - optimal tear streams according to Barkley and Motard[15] and Christensen and Rudd;[16] B - optimal tear streams according to Upadhye and Grens[17] and Motard and Westerberg.[18]).

many different tearing algorithms have been proposed over the years. However, all of these algorithms can be grouped in three different families, which rely on as many different underlying ideas:

- According to Barkley and Motard,[15] the tear streams should be selected as the smallest set of process streams that break all the recycles in the process flowsheet (Fig. 4).
- According to Christensen and Rudd,[16] the tear streams should be selected as the set of process streams that break all the recycles in the process flowsheet and minimize the size of the recycle problem (Fig. 4).
- According to Upadhye and Grens[17] and Motard and Westerberg,[18] the tear streams should be selected as the set of process streams that break all the recycles in the process flowsheet, under the additional requirement that the number of times every recycle is torn be minimum (Fig. 4).

Although none of these three alternatives is fully satisfactory, the best option depends on the numerical method, used to solve the recycle problem. More specifically, if we solve the recycle problem with Newton-type methods, then the second option represents the most appropriate choice from a computational perspective. On the other hand, if we solve the recycle problem with fixed-point algorithms, then the third option is the most suitable from a computational point of view (this tearing scheme maximizes the rate of convergence of fixed-point methods, when they are applied to Eq. 6). For computational (and historical) reasons, all the main software were originally designed as SM simulators. Hence, the first developers focused on the implementation of a single tearing algorithm that worked best in a SM environment. Therefore, most process simulation packages implement a single tearing algorithm, which often resembles the third, so the user's choices are limited to what is available in existing process simulation software. Due to the mathematical complexity of the tearing algorithm, we recommend a non expert user to operate the default method.

2.3.2 | Solution of the single unit operation models

A key step of all modular solution strategies for process simulation is the solution of the recycle problem (Eq. 6), which in turn requires repeated calculations of single unit operations. Since models of different types of unit operations exhibit unique features, dedicated numerical methods to solve each and every one of them have been developed over the years. However, all these numerical algorithms belong to either partitioning methods or simultaneous methods. Partitioning methods decompose unit operation models into blocks of equations, which are then solved iteratively, in a predefined sequence, until convergence criteria are met or the iterations diverge. Internally, they rely on fixed-point methods, thus exhibit the same numerical properties as this family of algorithms. Simultaneous methods do not decompose unit operation models but rather solve them as nonlinear systems. Since they apply Newton-type methods (possibly complemented with some parametric continuation approach), they exhibit similar numerical properties as these types of algorithms. Most process simulation packages implement simultaneous methods for unit operations but offer both partitioning and simultaneous alternatives for specific types of unit operations, e.g. distillation, absorption and stripping towers (for completeness, the most common partitioning solution scheme for separation towers is the inside-out method by Boston and Sullivan,[19] Saeger and Bishnoi[20] and Jelinek[21]). As a rule of thumb, we recommend partitioning methods for solving mildly non-ideal separations and simultaneous methods for solving highly non-ideal or reactive separations. In addition, simultaneous methods may be more appropriate for specific tasks, such as running case studies and/or performing flowsheet optimization, because of their higher computational efficiency. Additional recommendations on which numerical solution scheme to choose for specific simulation problems are usually available on the user guides of process simulation packages.

2.3.3 | Solution of the recycle problem

The last key component of any modular solution strategy for process flowsheets is the solution of the recycle problem (Eq. 6), which apply either fixed-point or Newton-type methods. In the 60s and 70s, we witnessed the birth of several new numerical algorithms, specifically designed to solve this problem. Nowadays, those that are still implemented in process simulation packages are the direct substitution method, the Wegstein method,[22] the dominant eigenvalue method,[23] the Newton-Raphson method and the Broyden method.[24] The rationale of each are:

- The direct substitution method is the simplest and oldest fixed-point algorithm, whose recursion matches Eq. 5 with all the relaxation factors $\boldsymbol{a}_{i,n} = 1$.
- The Wegstein method is an accelerated version of direct substitution, in which the relaxation factors $\boldsymbol{a}_{i,n}$ are computed by Eq. 7. We emphasize that Eq. 7 relies on the assumption that every variable of the recycle problem is independent of the others, so this numerical method may not perform satisfactorily when the Jacobian matrix of $f(\boldsymbol{x})$ (Eq. 6) is not diagonally dominant.
- The dominant eigenvalue method is an improved version of Wegstein, in which the relaxation factors $\boldsymbol{a}_{i,n}$ are computed by Eq 8. In this case, Eq 8 relies on the assumption that the Jacobian matrix of $f(\boldsymbol{x})$ (Eq. 6) has a dominant eigenvalue, so this numerical method may not perform satisfactorily when several variables of the recycle problem are equally important.
- The Newton-Raphson method is the simplest and oldest Newton-type scheme, whose recursion matches Eq. 2. It relies on the actual Jacobian of Eq. 6, so it may be more computationally efficient than Broyden when the recycle problem is a small-scale/medium-scale nonlinear system.
- The Broyden method is a Newton-type scheme, whose recursion matches Eq. 3. It relies on an estimate of the Jacobian of Eq. 6, computed by Eq. 9, so it may be more computationally efficient than Newton-Raphson when the recycle problem is a large-scale nonlinear system.

$$\boldsymbol{a}_{i,n} = \frac{1}{1 - \boldsymbol{s}_{i,n}} \wedge \boldsymbol{s}_{i,n} = \frac{f_i(\boldsymbol{x}_n) - f_i(\boldsymbol{x}_{n-1})}{\boldsymbol{x}_{i,n} - \boldsymbol{x}_{i,n-1}} \quad \forall i \in [1, N] \quad (7)$$

For Peer Review

$$\boldsymbol{a}_{i,n} = \frac{1}{1-\lambda_n^{MAX}} \wedge \lambda_n^{MAX} = \frac{\|\boldsymbol{x}_n - \boldsymbol{x}_{n-1}\|}{\|\boldsymbol{x}_{n-1} - \boldsymbol{x}_{n-2}\|} \forall i \in [1, N] \quad (8)$$

$$B_n = B_{n-1} + \frac{g(\boldsymbol{x}_n) - g(\boldsymbol{x}_{n-1}) - B_{n-1}\boldsymbol{d}_{n-1}}{\|\boldsymbol{d}_{n-1}\|^2}\boldsymbol{d}_{n-1}^T \wedge B_0 = J(\boldsymbol{x}_0) \quad (9)$$

All of these numerical approaches exhibit advantages and limitations, which make them more adequate for specific types of process flowsheets. Tab. 1 collects a general set of recommendations to help select a numerical alternative for most real-world simulation problems.

2.4 | Equation-oriented (EO) solution strategies

As opposed to modular solution approaches, equation-oriented solution strategies for process simulation do not attempt to simplify flowsheet calculations by exploiting the sequential nature of chemical processes. They construct the large-scale sparse system of nonlinear equations, which describes the process flowsheet of interest and includes unit operation models, process streams and design specifications, and solve it using dedicated numerical methods. The numerical schemes, suitable for flowsheet simulation, are Newton-type methods, combined with some merit function minimization strategy and, very often, some parametric continuation approach, which helps the sequence of iterations converge whenever equations are highly nonlinear, their Jacobian matrix is locally rank deficient, and/or a good initial guess cannot be computed.[6]

Regardless of any specific numerical details, equation-oriented solution strategies solve any type of process flowsheets, independently of their scale and topology (note that the performance of these solution schemes is unaffected by the number of recycles in the process flowsheet, provided that adequate computing power is available). However, they should be primarily used to solve process flowsheets, which contain many recycles, as modular solution approaches are more computationally efficient for simulation of process flowsheets with few recycles.

3 | APPLICATIONS

The program VoSViewer generated a bibliometric map of keywords generated from articles indexed by Web of Science Core Collection (WoS) from 2017 to April 2020.[25, 26] We queried the database 4 times with ASPEN +, ASPEN Plus, HYSYS, and PROII as topics, while simultaneously excluding forest. (To avoid work related to Aspen trees and forests rather than process simulation). We combined the four databases into one and eliminated all duplicate and ended up with 2582 articles. The Web of Science category energy & fuels had 930 articles while the chemical engineering category was a close second with 824 articles followed by thermodynamics (314 articles), environmental engineering (224), and green & sustainable science and technology (215).

We retained about 90 of the keywords that were cited most frequently in these articles but excluded *simulation*, *performance*, *column*, *behaviour*, and *growth*. VOSViewer grouped the keywords into 5 clusters with Aspen+ at the centre of the red cluster with the most articles (433) and the cluster with the most keywords at 29 (Figure 5). This cluster includes topics related to biomass (193 articles)— ethanol and bioethanol (132), lignocellulosics, and cellulose (64). The two other major topics relate to catalysis/kinetics (162) and techno-economic analysis/economics (227) and other subjects related to biomass.

Process simulation discriminates between process alternatives (Design—246 articles), which comprises modelling (187 articles) and determines the dimensions of unit operations like separation (126) including distillation (137 articles) (reactive and extractive), heat integration (33), and dehydration (26). Hung et al., for example, demonstrated that recovering dilute acetic acid water solutions (between 30 % to 70 % by mass of acid) with reactive distillation, amyl alchols among C_1–C_5 alcohols minimizes the total annual cost (TAC), without any stream pretreatment.[27] Pirola et al., simulated an extractive distillation column employing p-xylene as entrainer to maximize acetic acid recovery.[28] Xiao et al. dimensioned a three-column process to transesterify ethylene glycol with methyl acetate, regressing reaction kinetics parameters and optimized the design minimizing the TAC.[29] Wu et al. designed a dividing-wall column, together with its control strategy, for reformed gasoline.[30] Biodiesel and esterification are concepts less related to process simulation as they are far from the map center. This is because first- and second-generation biofuels interest is dropping in recent years in favour of other green technologies for the conversion of green house gases, such as CO_2 into fuels.[31–33] In fact, the magenta cluster centered around H_2 (227 articles) concentrates on gasification (biomass+steam with 239 articles), syngas (141), and reactors and fluidized beds (84).

CO_2 capture (193 articles), together with CO_2 (162), and CH_4 dominate the yellow cluster that includes combustion (87), absorption (82), and Hysys (66). This cluster spans a large area covering technology related to carbon capture and sequestration (CCS). Duhoux et al., for example, calculated the optimal pressure and flowrate of a pressurized fluidized bed combustion that sequestrates carbon dioxide via Calcium looping.[34] *Joule* has published the most cited article since 2019 (101 citations) that describes a plant to capture 1 Mt CO_2 per year with aqueous KOH sorbent and a calcium caustic recovery loop.[35] In fact, the article spans many of the keywords in

For Peer Review

TABLE 1 Numerical methods for flowsheet simulation.

Numerical method	Classification	Robustness	Efficiency	Recommendations
Direct substitution	Fixed-point	Low	Low	Neither reliable nor efficient Avoid its use.
Wegstein	Fixed-point	Medium	Low	Suitable for flowsheets that contain weakly interacting recycles[a].
Dominant eigenvalue	Fixed-point	Medium	Low	Applicable to process flowsheets with weakly and/or strongly interacting recycles[a,b]. Use when Wegstein fails
Newton-Raphson	Newton-like	Low	High	Fast but sensitive to initial guesses. Useful for flowsheet optimization and/or running case studies.
Broyden	Newton-like	Low	Medium	Equivalent to Newton-Raphson, more efficient for large-scale process flowsheets.

[a]: A set of recycles exhibits weak interactions when changes in the properties of any recycle stream induce small variations in the properties of all the other recycle streams.

[b]: A set of recycles exhibits strong interactions when changes in the properties of any recycle stream cause consistent variations in the properties of all the other recycle streams.

For Peer Review

the map—unit operations, capital cost estimates, and heat and mass balances with ASPEN+.

The Optimization (266 articles) cluster is connected to both Design and CO_2 capture (Figure 5). Many of the subjects relate to Process Intensification, which includes energy (228 articles) systems (215), energy/efficiency (127), and exergy (163).[36, 37] Dynamic simulation analyzes process start-up and shut downs, and closely connects to process design.[38]

Process simulation is an educational tool to demonstrate many facets of design.[39, 40] Steady-state simulation represents the plant at stationary conditions while dynamic simulation is required for the plant control strategies, for start up and showdown operations, and for operator training by plant virtualization.[41] The suggested approach for undergraduate students is steady state simulation,[42] while dynamic simulation is appropriate for graduate level courses.[43] For instance, the Universidad Complutense of Madrid, introduced steady state simulation software in the fourth year course of the chemical engineering degree,[44] proposing a case study on the reactive distillation for the production of MTBE. Simulation in chemical plants educational courses guides students through the principles of unit operations, and is a support in the development of a whole process.[45] For example, simulations reveal theoretical trends of processes, verify the kinetic and thermodynamics of reactions, and calculate fluid phase mixing and equilibria.[42] When proposing experimental activities in parallel with theoretical lectures and simulation, a creative environment that promotes team work is generated.[46]

Process simulation requires a solid theoretical background, as well as a basic knowledge of the computational methods the software are based on. For this reason, lectures on the simulation suite and capabilities are propaedeutic to understand and operate correctly.

Dynamic simulation is required for control strategies, for start up and showdown operations, and for operator training by plant virtualization.[47] The detailed representation of commercial control algorithms, the high-realistic models of several unit operations and the reproduction of the real behaviour of pipes, vessels, and valves are the basis of the most advanced dynamic simulation tools.[43] These options allow students to interact with a virtual plant and to operate instruments. There, the students engage at a practical level (starting up of a pump, sampling, reading gauges, etc.) and in plant management and control. This immersive simulation software combines augmented reality, computational dynamics and computer graphics.[48]

Augmented reality simulation examples concerning simulated chemical plants are rare, while robotics and physical sciences already developed commercial applications and university courses,[49] Nevertheless, new educational proposals are on-going, as for example Eye4Edu project at the University of Milano (Italy). Eye4Edu applies EYESIM software, from AVEVA, that proposes the immersive virtualization of a Crude Distillation Unit (CDU) plant with the representation of the detailed realistic 3D graphics of the whole plant

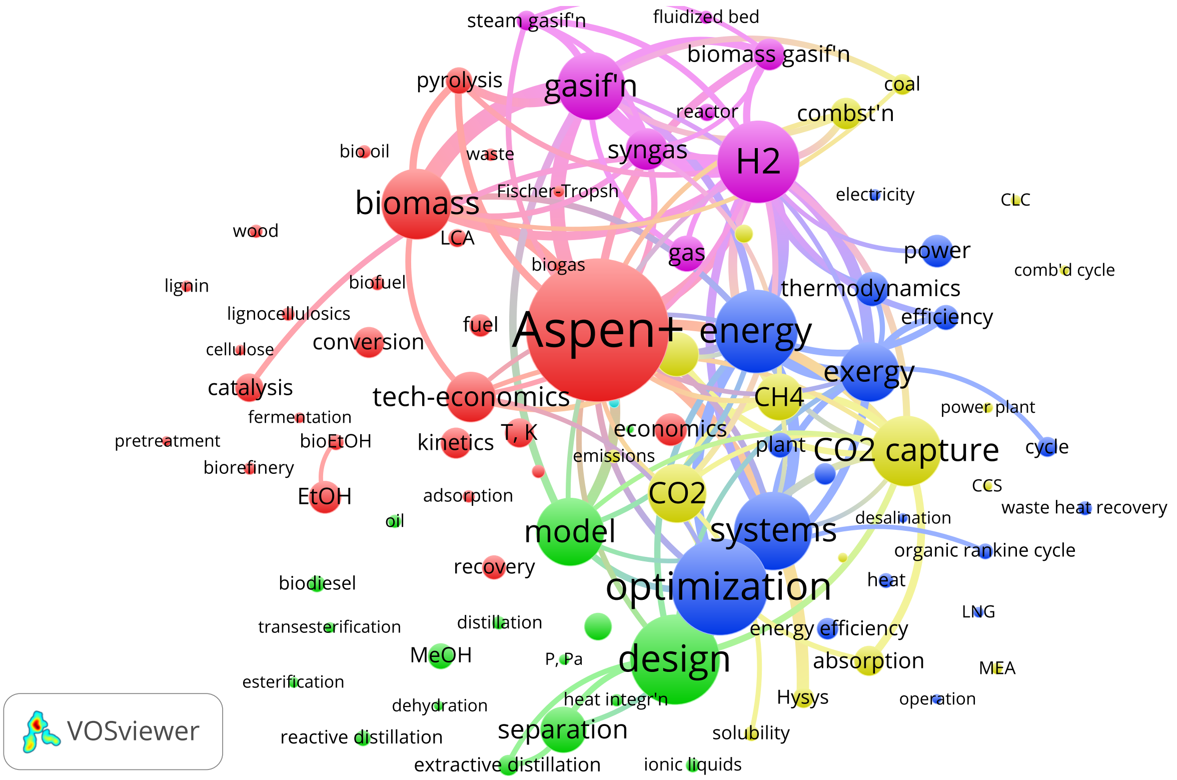

FIGURE 5 VOSViewer keyword bibliometric map based on keywords (Aspen+, Aspen plus, Hysys, and Pro/II) derived from articles that WoSof Science indexed from 2017 to April 2020.[25, 26] The text font size and diameter of the circles are proportional to the number of occurrences in the 2583 articles indexed during this time. Aspen+ (largest circle) appeared in 433 articles. The smallest circles appear in 26 articles.

(Figure 6).[50] DYNSIM software simulates the plant and supplies the physico-chemical properties. The main problems and limitations of the software in education are: 1) the imperfect reproduction of the reality, i.e. there is always some discrepancy compared to the real world. It is therefore impossible to recreate all the layouts of a real plant; 2) simulators are expensive and require maintenance and constant updates; and, 3) operators and teachers need training. Process simulators also train senior engineers and field operators.[51–53] Commercial software simulate mostly reactions, separations, design heat integration and offer a variety of ideal reactors.[54, 55] Depending on the field of application, the software is integrated in the design of a new process, in the simulation of an existing plant, or both.[56] For instance, if we simulate and benchmark the energy consumption of an existing plant, we increase its overall efficiency and save money. In 2017 and 2018, Panjeshahi et al. demonstrated the efficacy of the process integration in the cement[57] and the petrochemical industry,[58] with a total duty saving of 24 % and 14 %, respectively. Cold and hot utilities consumption, energy requirements and emissions of oil & gas,[59, 60] electric power,[61] biofuels,[62] chemical, pharmaceutical and urban systems[63] existing plants can all be reduced with a retrofit based on a simulated process. Optimize a process, instead, requires setting economic, process or environmental objective functions to satisfy. Commercial software, such as AspenPlus, Hysys, PRO II or SuperPro Designer, are designed to work with a fully defined problem (sum of degrees of freedom equal to 0).

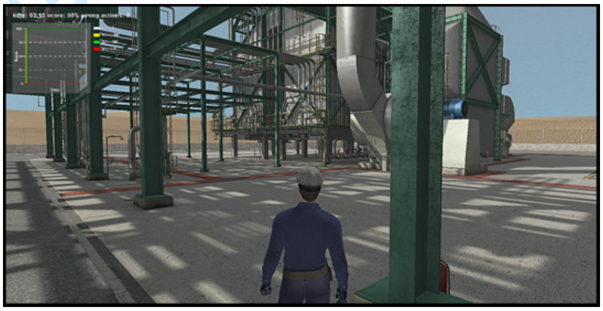

FIGURE 6 Operator Training Simulators (OTS) are an educative tool for master and undergraduate students. A dynamic simulation of the plant calculates the consequences of each operator's actions and the student learns through experience.

3.1 | Cost estimation

The log of the investment (log I) of a process unit and even an entire plant is linearly correlated with the log of its capacity (log Q). The slope of the curve depends on the equipment type, specifically on its characteristic dimension,[64] and it varies from 0.33 to 1.0 for equipment and from 0.38 to 0.90 for plants, averaging to a value of 0.6[65] (Eq. 10).

$$I_2 = I_1 \left(\frac{Q_2}{Q_1}\right)^{0.6} \tag{10}$$

For modular units, the learning elasticity is proportional to this exponent and the cost of these units decreases with multiple units—learning/experience.[66] Lang elaborated a simplified method to calculate the total installed cost of equipment based on factors that vary according to the type of plant (solid handling, solid-fluid, or fluids).[67–69]

A more detailed approach, proposed by Guthrie, estimates the bare module cost of a unit, C_{BM}, based on a characteristic variable, A, (volume for a reactor or a column, surface for an heat exchanger, etc.) corrected with pressure coefficients, C_i and material coefficients, B_i (Eq. 11).[70, 71]

$$\begin{cases} C_{BM} = C_p^0(B_1 + B_2 F_M F_P) \\ \log_{10} C_p^0 = K_1 + K_2 \log_{10} A + K_3 \left(\log_{10} A\right)^2 \\ \log_{10} F_P = C_1 + C_2 \log_{10} P + C_3 \left(\log_{10} P\right)^2 \end{cases} \tag{11}$$

F_M the material factor, that is unity for carbon steel and > 1 for other materials, and F_P is a pressure factor. In the case of vessels, a different equation calculates $F_{P,vessel}$ (Eq. 12)

$$\begin{cases} F_{P,vessel} = \dfrac{\dfrac{(P+1)D}{2(850-0.6(P+1))} + 0.00315}{0.0063} \\ F_{P,vessel} = 1 \quad \text{if} \quad t_{vassel} < 6.3\,\text{mm} \\ F_{P,vessel} = 1.25 \quad \text{if} \quad P < -0.05\,\text{MPag} \end{cases} \tag{12}$$

which is valid for a corrosion allowance of 3.15 mm.

Ulrich and Vasudevan improved the method of Guthrie and extended his correlation to 58 unit operations of the chemical industry and updated the correlations with a larger database.[72, 73] All the estimations are then actualized by an inflation index like the Marshall & Swift equipment Cost Index, the Chemical Engineering Plant Cost Index (CEPCI), the Engineering News-Record (ENR) construction Index, or the Nelson-Farrar Refinery Construction cost Index (NFRCI), where CEPCI and NFRCI refer to the entire plant and the others to either plant or equipment.[74] Simulation packages have tools to estimate a units' costs, that compute a combination of the methods reported with vendors' quotes/estimates.[75] They prefer to implement algorithm-based methods (Guthrie, Ulrich, Seider[76]) rather then cost-to-capacity or the Lang-based methods. For instance, Aspen Capital Cost Estimator (ACCE)—formerly APEA, the in-house Aspen estimation software, couples design and equipment cost modules, with proprietary time-proven, industry-based direct and indirect factors based on company project history, vendor quotes. These factors are updated regularly.[76] Industry and academia developed dozens of integrated and stand-alone software to estimate cost. Cleopatra and ACCE are the built-in software of PRO/II and AspenOne, respectively. Another example is the built-in Capital Cost Estimator of the CAPE-OPEN project DWSIM. EstPro is a stand-alone process plant cost estimation package from Gulf Production while CapCost, CCEP and DFP are included with books.[74, 76, 77] EconExpert is web-based software. Even software heavily based on updated vendor quotes have errors of at least 25 %.[75] We recommend to operate with one model (or software) to evaluate different plant designs to generate comparable results.

3.2 | Optimization

Most process synthesis problems have multiple feasible solutions. Quantitative optimization techniques select the best of these solutions, which corresponds to a process configuration that maximizes a problem-specific measure of process performance. Any process synthesis problem can be reformulated as an equivalent optimization problem (Eq. 13) composed of three principal components: an objective function, $f(\boldsymbol{x}, \boldsymbol{y})$—a problem-specific measure of process performance, some decision variables, continuous $\boldsymbol{X}$ (process variables), or binary $\boldsymbol{y}$ (they specify the presence or absence of certain process units), and a set of constraints, inequality $g(\boldsymbol{x}, \boldsymbol{y})$ (e.g., minimum product purity, maximum allowable equipment cost, environmental emission limits) and equality $h(\boldsymbol{x}, \boldsymbol{y})$ (e.g., mass, momentum and energy balances, mass and heat transfer correlations, and phase equilibria). The numerical solution of this optimization problem requires discrete or continuous solvers.

$$\begin{cases} \text{Minimize} \quad f(\boldsymbol{x}, \boldsymbol{y}) \\ \text{Subject} \quad \text{to} \quad g(\boldsymbol{x}, \boldsymbol{y}) \le 0, \quad h(\boldsymbol{x}, \boldsymbol{y}) = 0 \\ \boldsymbol{x} \in \boldsymbol{X}, y \in \{0, 1\} \end{cases} \tag{13}$$

Examples of common objective functions include total cost, profit, energy expenditure, exergy loss, environmental and/or social impact, and the error over time of a control model. When two or more conflicting objective functions must be maximized/minimized simultaneously, a dedicated optimization method solve this multi-objective optimization problem. For example, Patel and Padhiyar[78] solved a bio-reactor design problem, in which they simultaneously minimized the batch time and maximized the process yield.

Additionally, Eq. 13 may have multiple solutions, called local optima, and finding the best of these, i.e. the global optimum, is essential. In this case, the user selects special

For Peer Review

types of optimization algorithms that offer global optimality guarantees:

- Linear problems (LP): linear objective function and constraints, and continuous decision variables,
- Mixed-integer linear problem (MILP): linear objective function and constraints, and both discrete and continuous decision variables,
- Non-linear problems (NLP); non-linear objective function and/or constraints, and continuous decision variables, and
- Mixed-integer non-linear problems (MINLP): non-linear objective function and/or constraints, and both discrete and continuous decision variables.

Scheduling, supply-chain and planning problems are usually LPs or MILPs. NLPs and MINLPs characterize dynamic optimization, design and process synthesis problems. Gradient-based and derivative-free algorithms solve NLPs and MINLPs. Both the algorithms are iterative: they start from an user-supplied initial guess and progressively move towards a solution of the optimization problem (a set of optimal values of the decision variables). Gradient-based methods rely on the values and the derivatives of constraints and objective functions while derivative-free methods rely on the value of the functions. Gradient-based algorithms are suitable for mildly non-linear optimization problems with smooth objective functions and constraints, given that the initial estimates are coherent. Derivative-free methods are appropriate for highly non-linear small-scale optimization problems with both smooth and non-smooth objective function and constraints, even if a good initial guess is unknown. These optimization algorithms are partially complementary, which is the reason why process simulation packages implement a few gradient-based and derivative-free methods, and let the user excogitate on the one to select.

The LP simplex method for LPs,[79] sequential quadratic programming (SQP) for NLPs,[80] and the branch & bound (BB) method for MINLPs are examples of gradient-based methods.[81] SQP is a quasi-Newton algorithm that is computationally efficient but it relies on first and second order derivatives of the objective function and constraints of the optimization problem. For this reason, SQP solves smooth problems best such as LPs and NLPs (Aspen Plus and PRO/II feature SQP as default optimization method). Additionally, SQP does not guarantee convergence to the global optimum.

Derivative-free optimization strategies include BOX, which is a basic implementation of the Nelder-mead simplex algorithm,[82] and COMPLEX, whose implementation details are property of Aspen HYSYS and Aspen Plus. These built-in algorithms handle simple optimization problems with a single objective function, provided that a decent initial guess is available.[83, 84] However, complex or multi-objective optimization problems may require external optimization packages (e.g., CPLEX and XPRESS for solving LPs and MILPs, CONOPT, IPOPT and KNITRO for solving NLPs, and DICOPT, BARON and ANTIGONE for solving MINLPs), interfaced with the simulation flowsheet. For instance, Galli et al.[85] applied Monte Carlo optimization of operating costs and environmental impact of a plant for producing oxygen-enriched air, simulated with PRO/II). Quiroz-Ramírez[86, 87] optimized a bio-butanol plant with AspenPlus, MATLAB, and a home-made VBA script. Eslick et al.[88] optimized an amine absorption process, with Excel, Aspen HYSYS, and the solver NSGAII. Ponce-Ortega et al.[84] developed a procedure to tune any commercial simulator to any multi-optimization algorithm with MS Excel. In fact, they implemented a hybrid stochastic algorithm called I-MODE to maximize the gross annual profit and to minimize the CO_2 emissions of a power and a biodiesel plant.[89] Interfacing commercial simulation packages with external optimization tools is a considerable task with respect to computational power, and time, so we recommend this technique only as a last resort.

3.3 | Energy Integration Intensification

Since 1970,[90] pinch analysis has guided the placement of heat exchanges and other unit operations (distillation, for example) to minimize energy consumption.[9] We visualize each non-reactive stream by a heat content H (kW)—temperature couple. When a differential heat flow dQ is added to a stream, it increases its enthalpy H by $C_P dT$:

$$Q = \int_{T_1}^{T_2} C_P \, \mathrm{d}T = C_P(T_2 - T_1) = \Delta H \qquad (14)$$

where T_2 and T_1 are the target and the supply temperature, respectively. In this way, plotting the temperature in terms of the heat content, we have, for each stream, a curve whose slope is $1/C_P$. For a single couple of hot-cold streams this approach is of little use, because to a k hot load increase, corresponds a k cold required duty increase.[91] However, the graphical representation comes in handy when there are several streams involved. In this case:

- **First** plot all the streams separately on a T/H graph. Since we are interested in the ΔH, we only have to place the stream on the right interval of the y axes.
- **Then** add up all the cold and heat duties separately. Since the slope of each stream represents its C_P, whenever two or more streams share a temperature interval $T_1 - T_2$, the heat available in this interval will be $(C_{P,i} +$

For Peer Review

$C_{P,ii} + \dots C_{P,n})(T_1 - T_2)$. The outcome of this procedure are the hot and cold composite curves.

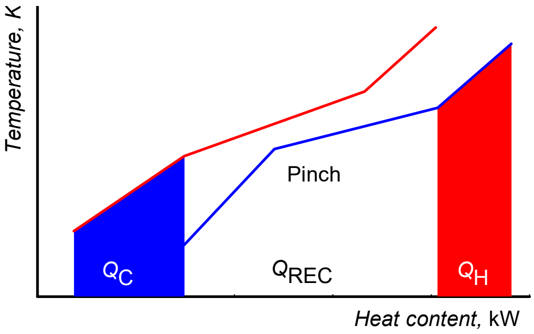

FIGURE 7 Example of hot (red) and cold (blue) composite curves. The blue area is the minimum cooling duty required Q_c and the red area is the minimum heating duty required Q_h.

For Peer Review

When we plot these together, we identify (Fig.7):

- The minimum cooling duty required Q_C.
- The minimum heating duty required Q_H.
- The recovered duty where the curves overlap Q_{REC}.
- The point of closest approach, the pinch point.[90].

This means that as long as the heat exchange across the pinch operates at the corresponding minimum ΔT, it is possible to design an heat exchange network (HEN) who will recover Q_{REC}. The only external heat for the process will only be Q_H and Q_C. Lindhoff et al.[92] expanded the graphical concept of the composite curve, to the *Problem Table* method. This latter method, algebraically splits the process in a cascade of temperature intervals. To do so, we adjust the hot or cold stream temperatures of a negative or positive fraction of ΔT_{pinch}. For instance, we either shift the hot streams by -$0.5\Delta T_{\text{pinch}}$ and the cold streams by $+0.5\Delta T_{\text{pinch}}$, or the hot streams by $-\Delta T_{\text{pinch}}$ and leave the cold strea,s untouched. Setting the temperatures this way guarantees that each interval will either have a positive or a negative duty balance. Foreach interval i we will therefore have a balance:

$$\Delta H_i = (T_i - T_{i+1})(\Sigma C_{\text{P,h}} - \Sigma C_{\text{P,c}})_i \quad (15)$$

When we sum up the ΔH for all the intervals, we obtain a positive and a negative heat residual. The only way to close the energy balance is to introduce a cold and a hot utility. We have to supply a Q_C and Q_H at a thermal level lower and higher of the lowest and highest interval of the cascade. That is, if the highest interval is at $T = 160\,°\text{C}$, we include a hot utility above 160 °C. Brown et al.,[93] demonstrated how pinch technology achieved an average of 25 % cut on the site energy bills, equivalent to 30 million pounds in 1989, for all the main sectors in chemical engineering. Although other methods like the composite curve method are equivalent,[76, 94] the inherent modularity of the Problem Table method makes it perfect to implement it in process simulators.

Process simulators perform pinch point analysis (PPA) to design an optimal heat exchange network (HEN), minimize the energy requirement of the process and the utilities consumption.[54] Several software packages on the open market incorporate a PPA section such as HYSIM, HEXTRAN, ADVENT by Aspen Tech, TARGET or, more recently, FI^{2}EPI.[95]. Simulators couple PPA analysis with a UA heat exchange model, in this way they work on the design of the network outside the simulation environment. They share the possibility of importing the H&M balance results from outside, and the presence of at least a manual and an automatic design. However, we recommend completing individual PPA before integrating them all into the entire network. In fact, the automatic mode will achieve the target (minimum number of heat exchangers or maximum energy saving), regardless the process engineering of the system. For instance, it might suggest you to exchange between stream 1 and stream 1000 of the plant, not considering the physical distance. Or it might suggest to split one stream into five sub-streams, to maximize its heat exchange. Or, based on what you are looking for, it might propose uneconomical heat exchangers. On the other hand, this software will not suggest you where to locate an equipment with respect to the pinch.[9, 76] Furthermore, energy intensification works as long as the plant operates at steady-state. However, while the steady-state defines the baseline operation,[96] in real life a plant alternates between steady-state and unsteady state (start-up, shut-down, feed composition variations, for example).[97–99] For this reason, we recommend understanding the dynamics of all the operations in the simulation before integrating energy.

3.4 | Process simulation as research tool

Process simulators started as academical tools but soon thereafter industry adopted this software for design. Now this tool is returning to academia and research: Saidi and Kadkhodayan[100] integrated an experimental Taguchi optimization method to a process simulation to bridge industrial operation with laboratory scale experiments. Jafari simulated a circulating fluidized bed with a sequential modular approach and showed that it could be added to a commercial simulation.[101] Commercial software developers are expanding their offering with non conventional components, such as electrolytes, solids, or plastic[56]. Process simulators support the experiments and approximate thermodynamic parameters with equations of state for those parameters difficult to measure. Shen et al.[102] studied a primary cooler for a Coke Oven Gas LiBr heat pump. They relied on simulation results to estimate operating parameters they were unable to measure experimentally to build a refrigerator pilot plant. They retroactively validated the simulation with pilot plant data and reported an error of 0 % to 10 % depending on the parameter. Process simulators are also tools to calculate thermodynamic properties—enthalpy, entropy, Gibb's free energies, densities, viscosity—as input to estimate reaction kinetics, heat duties, and phase changes. analysis of a reaction. For example, to define an experimental plan to study methane partial oxidation, we estimate the theoretical bound on conversion and selectivity as a function of temperature and pressure, then determine the relationship between coking and oxygen partial pressures. These data define the boundaries of the plan. PRO/II simulated a Gibbs reactor (RGibbs) with the Peng-Robinson equation of state. A case study varied the O_2/CH_4 ratio in the range 0.2 to 1.0 and the temperature of the reactor from 1000 K to 1200 K (Figure 8).

For Peer Review

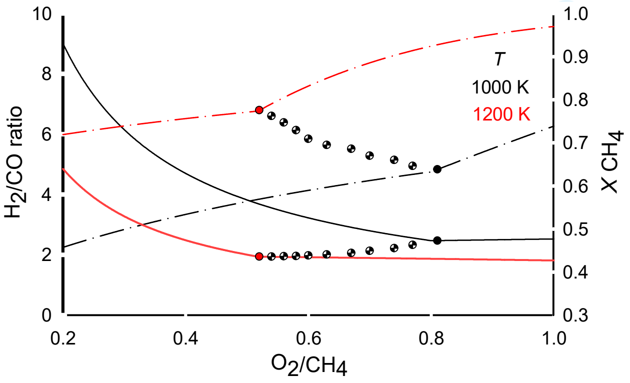

FIGURE 8 Thermodynamic (maximum) methane conversion (dotted lines) and obtainable H_2/CO ratio (full lines) for the catalytic partial oxidation of methane at 1000 K (black lines) and 1200 K (red lines). The simulator also identifies at which ratio coke starts to form (points). We calculated the H_2/CO ratio at which coke forms every 20 K. The simulations took less than 8 min to complete.

4 | UNCERTAINTY

4.1 | Limitations

Commercial software licenses are expensive and to simulate non-standard operation often additional code is required. The commercial software models and algorithms are robust, and most of the errors come from user-added inputs. Integrating custom code into process simulators is cumbersome and takes longer than writing a program from scratch.

To set up a simulation file, the following procedure applies:

- Define the units of measure, according to a predefined set (metric, english or SI) or to a customized set,
- Select the components (from a databank),
- Choose a thermodynamic model,
- Check the simulation tolerances, and
- Build the flowsheet. In case of recycles see Section 2

Errors compound as the simulations proceeds from one step of the process to the next. Simulators have no error assessment capability.

4.2 | Sources of error

4.2.1 | Units of measure (UOM)

One of the most frequent error source is incorrect data entry, i.e. the human error. Even though most of the commercial simulation suites have a window dedicated to the selection of UOMs, it is possible to change these inside all unit operations. For example, if the ENGLISH system is set as default, but for a specific distillation application the user knows that the distillate flowrate is 100 kmol h^{-1}, inserting a value of 100 without correcting the UOM will result in setting a distillate flow of 100 klbmol h^{-1}, resulting in a error of 1/2.2. The most likely error is the simulation will not converge to a solution.

Another common mistake is when users define units of measure in the reaction kinetics window. The user inputs the main kinetic parameters—Arrhenius constants, A_{pre}, activation energies, E_a,and reaction orders, ϵ—for a kinetic model or by defining a pseudo-homogeneous kinetic scheme (Eq. 16):

$$r = A_{pre} \exp\left[-\frac{E_a}{R}\left(\frac{1}{T}-\frac{1}{T_o}\right)\right]\cdot T^n \cdot \prod a_i^{\epsilon} \qquad (16)$$

where a the activity of reagents and products (calculated as the product concentration, molar fraction or according to a thermodynamic model).

PRO/II permits the user to supply the values of these parameters and the volume and the pressure UOMs, as the r is always defined as (mol Volume^{-1} Time^{-1}). Also, the value of the gas constant R is 8.314 J mol^{-1} K^{-1}. Therefore, the UOMs of pressure and volume in the reaction window must be chosen to give J when multiplied, i.e. or kPa and L, or Pa and m^3, respectively.

On the other hand, when defining a kinetic procedure, the user writes a FORTRAN or pseudo-FORTRAN code and calculates the system of differential equation that the software will integrate using a numerical method (typically Runge-Kutta). In this case, the user should provide the proper rate expression UOM to the solver, in mol Volume^{-1} Time^{-1}.

4.2.2 | Components selection

The selection of components means, for the software, to import all the pure component properties, the fixed (molecular weight, acentric factor, normal boiling point, critical point, Van der Waals area and volume, etc.) and the temperature dependent properties (density, vapor pressure, viscosity, heat capacity, etc.).

We always recommend to employ components present in the software database, as these are usually updated with one or more methods to estimate the component's properties. Besides, a good practice is to operate with less than 40 pure compounds per simulation.

In the case of new components (those not yet in any database), all the physical properties should be provided as well; we recommend to retrieve the physical and thermodynamic properties on databases such as NIST chemistry WebBook[103] or Knovel.[104]

There are dozens of equation to estimate thermodynamic properties.[105] This reference covers also electrolytes systems and solids solubilities. We recommend to pay particular attention while operating with new materials and processes such as:

- Green and biological processes,
- Molten organic and inorganic salts
- Nonconventional solids (i.g. solid with unknown chemical formula)
- Electrolytes
- Novel materials (nanomaterials, composite materials, copolymers, etc.)

Commercial software as ASPEN, PRO/II and ProSim estimate new component's properties from UNIFAC group contribution methods.[106] In this case, the flaws of the simulators rely in the estimation of the physical properties, which are either imprecise,[107] or incomplete.

4.2.3 | Thermodynamics

One of the most critical points is the selection of a proper thermodynamic method to calculate mixture properties (excess Gibbs free energy or fugacity). Analyzing the system (Fig. 9) leads to the selection of a $\gamma - \phi$ or a $\phi - \phi$ approach. However, this is a simplified scheme as each component mixture may show different non-idealities. For example, the system water and acetic acid (fully miscible, condensed phase and no supercritical gases dissolved) should be treated with an activity coefficient model. However, with UNIQUAC equations and the binary parameters available in PRO/II database only, the software calculates an non-existent azeotrope (Fig. 10). Only adding a correction for the gas phase association of acetic acid (Hyden-O'Connel equation[110]) or regressing with a robust method experimental data led to a sound simulation.[28] Therefore, we always recommend a literature analysis of the phase equilibria of the systems involved. Gmehling et al.[111] published a book regarding chemical thermodynamics and process simulation. Gani and O'Connel[108] created a decision matrix to select proper physical methods and thermodynamic models, depending on the unit operation simulated and the operative parameters.

4.2.4 | Tolerances

Similar to any other iterative calculation algorithm, the solver of a process simulator needs specific tolerances to terminate the calculations. Generally the default parameters (1×10^{-3} for temperature and pressure —relative), 1×10^{-6} for compositions (absolute), and 1×10^{-4} for duties—relative) are sufficient to guarantee the robustness of the termination criteria.

4.2.5 | Flowsheet

Any unit operation requires specifications to close the mass and energy balances. Generally these specifications are temperatures, pressures, or material flows (like the flow of distillate and residue for a distillation column) but any unit may have peculiar specifications (reactor dimensions for a plug flow or time for a batch reactor, reflux ratio for a distillation column, etc.). Converging to a viable solution depends on setting these specifications correctly. Most of the error in this case are related to the unit of measures of the specifications or to gross errors such as setting a too high reflux ratio or a distillate flow higher than the feed flow rate of the column.

Some specifications make the solver's life harder. If the simulation never converges to a solution, the user should replace some of them with equivalent specifications (e.g, instead of decreasing the molar fraction of impurities in the distillate, increase the reflux ratio until the composition reaches the desired value). Another common mistake is to set the composition of distillate or residue to a value impossible to obtain with the number of theoretical trays specified (such as imposing a distillate or residue purity higher than the minima/maxima azeotropic compositions) for distillations, or imposing by mistake to concentrate the heavy component in the distillate rather than in the residue stream.

If the user imposes extreme constraints (i.e. a separation in which the molar fraction of impurities in the product is less than 1 ppm), then they should start with a more relaxed constraint (±5 %) and tighten it up gradually.

For Peer Review

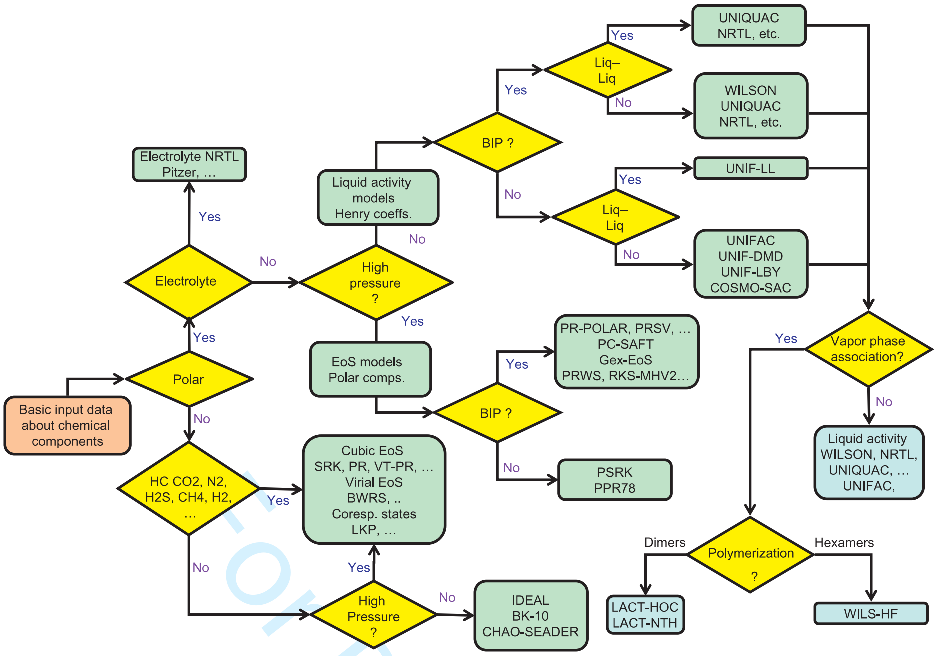

FIGURE 9 Simplified decisional flow diagram for the selection of a thermodynamic method (Tab 2).[108, 109]. *Reprinted from Computer Aided Chemical Engineering, Volume 13, Second Edition—Integrated design and simulation of chemical processes, Alexandre C. Dimian, Costin S. Bildea, Anton A. Kiss, Chapter 6: Phase equilibria, 248, Copyright (2020), with permission from Elsevier.*

TABLE 2 Most employed thermodynamic models to calculate components' fugacities or activity coefficients.

Model	Advantages	Limitations
	Equation of State (EoS)	
IDEAL	Good for mixtures of similar fluids at low P and T	$P < 0.3$ MPa
Chao-Seader[112]	Generalized correlation for hydrocarbon mixture	$P < 14$ MPa, $T < 260\,°C$ Dissolved gas $< 20\,\%$ by mole
Lee-Kesler[113]	Correlation for $\bar{H}$, $\bar{S}$ and $\bar{\rho}$	Not good for highly polar mixtures
Redlich-Kwong[114]	Calculates hydrocarbons VLE accurately	Not good for polar compounds
Peng-Robinson[115]	Better than SRK with polar components	Worse for hydrocarbonsmixtures
UNIWAALS[116]	Predictive method	Components' liquid volumes required
	Activity coefficients	
Wilson[117]	Polar or associating components	Does not calculate LLE
NRTL[118]	Strongly nonideal mixtures	Very different molecules dimensions
UNIQUAC[119]	Very good with most of systems	No electrolytes
UNIFAC[120]	Predictive method	$T << T_C$, $P < 8.5$ MPa

4.2.6 Initialization

All numerical methods for the resolution of unit operations require initial values to start iterating. However, all commercial packages possess several initialization methods for each

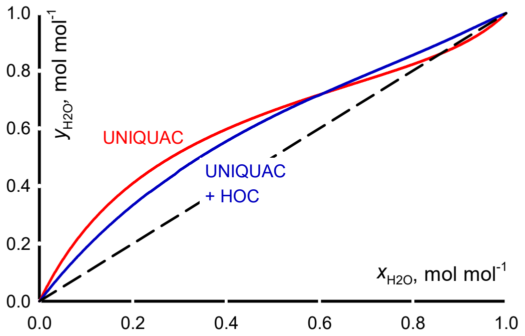

FIGURE 10 The correct selection of a thermodynamic method (blue line) avoids the calculation of non-existent azeotropes (red line) for the binary water-acetic acid system, $P = 101.325\,\text{kPa}$. UNIQUAC parameters: $a_{ij} = -118.309$, $b_{ij} = -0.124\,345\,\text{K}^{-1}$, $a_{ji} = 402.126$ and $b_{ji} = -0.355\,725\,\text{K}^{-1}$.

unit operation, and these algorithms vary from one unit to another. For an exhaustive description of each method consult the software manual.

For complex simulations, involving several units and recycling streams, we recommend choosing initial conditions judiciously, or to estimate them based on shortcut calculations or literature data, to facilitate the algorithm convergence to the solution. Poor initial guesses will increase the number of iterations unnecessarily (if the simulation converges at all)..

To initiate the initial guess for tear streams, a good strategy is to simulate the flowsheet with no recycle, and then connect the recycles, so that the software stores after the first simulation the initial values of flow rate, temperature, pressure, and compositions of these streams.

4.3 | Sources of error in the EO Approach

While thermodynamic, component selection, and flowsheets are common for every computational architecture, the EO is intrinsically free of any sources of error related to iteration. However, since the EO solves all the units at the same time, the initial values are essential. As a rule of thumb, we recommend to first run the simulation in a SM environment, and then implement the SM outcome as initial value for the EO. This doesn't mean that the SM needs to converge, but at least each block has to be solved once. The closer the SM solution is to the real solution, the easier it will be for the EO to converge without error. While this initialization technique is the one currently present in commercial simulators, new approaches such as the *graphic approach*,[121] or the *pseudo-transient*,[122] are recently emerging.

5 | CONCLUSIONS

Engineers design, control, optimize, retrofit, and estimate the cost of a plants and equipment with process simulators, which calculate thermodynamic properties, model vapor- and/or liquid-liquid equilibria of mixtures and close mass and energy balances. Process simulation will become ever more present in a chemical engineer's arsenal of tools to address society's desire for sustainable products that minimize waste and maximize recyclability as the mathematical methods and computational power improve such that any PC can run these multi-objective optimization problems. The feedstocks of the future will become more heterogeneous (bio-based, waste-based, for example) and because of this complexity, new models, unit operations and thermodynamic correlations will be needed. Quantification of the uncertainty on model predictions is another key aspect that should be further improved in future simulators as well as adding stochastic optimization capabilities to these tools.

Prospectives for future innovations include:

- generating kinetics databases (including packages like Chemkin;[123]
- handling solids and other unconventional materials (plastic mixtures, urban waste, sludge, and poorly characterized complex mixtures);
- better dynamic simulators that consider the time scale of reaction versus hydrodynamics contribution;
- new operating functions for process intensification[124, 125] of current processes;
- more reliable safety assessments that autiomatically identify hazardous operating conditions (HAZOP automatic calculation of lower and upper explosion limits,[126] identification of risk zones, improving the hazard identification when streams mix);
- integrating computational fluid dynamics;
- dynamically query other databases on the internet (NIST);
- integrating artificial intelligence and artificial neural network algorithms for control and optimization;
- improving the thermodynamic packages for new molecules (pharma and green chemistry);
- improving the predictive control of unit operations;
- expanding virtual reality that reproduces realistic plant dimensions; and,

For Peer Review

- improving the integration between home-made code and process simulators.

Also, the integration of multiobject optimization on performance key indicators not only based on economic parameters, such as the CO_2 equivalent or the water footprint of a process (environmental impact) or the hazard and operability analysis, which at the moment is feasable via open platform communication.[127].

As stated by John E. Coon et al. in 1998 "*It is the **responsibility** of the engineer to validate the quality of process design, troubleshoot, and optimization results from simulation tools*" and that wrong answers are "*the results of getting the correct answer to a poorly chosen question*".[128]

For Peer Review

NOMENCLATURE

A	Characteristic variable of an equipment
$\boldsymbol{a}$	Optimal step length
ACCE	Aspen capital cost estimator
A_{pre}	Arrhenius pre-exponential factor
a_i	Activity of component i
$B()$	Jacobian matrix estimation
B_i	Material coefficients
C_{BM}	Cost of bare module
CCS	Carbon capture and sequestration
CDU	Crude distillation unit
CEPCI	Chemical Engineering Plant Cost Index
C_i	Pressure coefficients
C_{P}	Specific heat
D	Diameter of a vessel
$\boldsymbol{d_n}$	search direction vector
E_{a}	Reaction activation energy
ENR	Engineering news-record
EO	Equation oriented approach
EoS	Equation of State
F_{M}	Material factor
F_{P}	Pressure factor
$\bar{H}_i$	Enthalpy of component i in mixture
HOC	Hyden-O'Connel
I	Investment
$J()$	Jacobian matrix
K_i	Cost coefficient of an equipment
LLE	Liquid-liquid equilibrium
LP	Linear problem
MILP	Mixed integer liner problem
MINLP	Mixed integer non-liner problem
MTBE	Methyl-tertbutil ether
NFRCI	Nelson-Farrar Refinery Construction cost Index
NLP	Non-linear problem
OTS	Operator training simulator
P	Pressure
PPA	Pinch point analysis
PR	Peng-Robinson
Q	Plant capacity
Q_{c}	Minimum cooling duty
Q_{h}	Minimum heating duty
Q_{rec}	Recovered duty
R	Gas constant
r	reaction rate
$\bar{S}_i$	Entropy of component i in mixture
SM	Sequential modular approach
SQP	Successive quadratic programming
SRK	Soave-Redlich-Kwong
T	Temperature
T_{C}	Critical temperature
t_{vessel}	Vessel wall thickness
TAC	Total annual costs
UOMs	Unit of measures
VLE	Vapor-liquid equilibrium
$\boldsymbol{x_n}$	solution vector at the iteration n
$\boldsymbol{x_s}$	solution vector

Greek Letters

γ	activity coefficient
ϵ	reaction order
ϕ	fugacity
$\bar{\rho}_i$	Density of component i in mixture

ACKNOWLEDGEMENTS

The authors aknowledge professor Rahmat Sotudeh Gharebagh for his precious comments and suggestions, and the fruitful discussions.

References

[1] J. J. McKetta, J. J. McKetta, ed., Encyclopedia of Chemical Processing and Design, CRC Press, New York **1980**, pp. 150 – 458.

[2] D. C. Y. Foo, N. Chemmangattuvalappil, D. K. Ng, R. Elyas, C.-L. Chen, R. D. Elms, H.-Y. Lee, I.-L. Chien, S. Chong, C. H. Chong, Chemical Engineering Process Simulation, Elsevier, Amsterdam, Netherlands **2017**.

[3] G. S. Patience, *Can. J. Chem. Eng*. **2018**, *96*, 2312.

[4] A. W. Westerberg, H. P. Hutchinson, R. L. Motard, P. Winter, Process Flowsheeting, Cambridge Universities Press, Cambridge, UK **1979**.

[5] R. S. Mah, R. S. Mah, ed., Chemical Process Structures and Information Flows, Butterworth-Heinemann **1990**, pp. 125 – 183.

[6] G. Buzzi-Ferraris, F. Manenti, Nonlinear Systems and Optimization for the Chemical Engineer, Wiley-VCH, Weinheim, Germany **2013**.

[7] Aspen Plus 2004.1 Getting Started Using Equation Oriented Modeling **2004**. URL https://sites.ualberta.ca/CMENG/che312/F06ChE416/HysysDocs/AspenPlus20041GettingStartedEOModeling

[8] H.-S. Chen, M. Stadtherr, *AICHE journal* **1985**, *31*, 1843.

[9] A. C. Dimian, C. S. Bildea, A. A. Kiss, Applications in Design and Simulation of Sustainable Chemical Processes, Elsevier **2019**.

[10] R. Raoni, A. R. Secchi, E. C. Biscaia Jr, *Comput. & Chem. Eng*. **2017**, *96*, 169.

[11] F. Zhao, X. Chen, L. Zhu, *AIChE J*. **2017**, *63*, 2764.

[12] H. A. J. Watson, Robust simulation and optimization methods for natural gas liquefaction processes, Ph.D. thesis, Massachusetts Institute of Technology **2018**.

[13] D. Himmelblau, *Chem. Eng. Sci*. **1967**, *22*, 883 .

[14] H. Weinblatt, *J. ACM* **1972**, *19*, 43 .

[15] R. Barkley, R. Motard, *The Chem. Eng. J*. **1972**, *3*, 265 , an International Journal of Research and Development.

[16] J. H. Christensen, D. F. Rudd, *AIChE J*. **1969**, *15*, 94.

[17] R. S. Upadhye, E. A. Grens II, *AIChE J*. **1975**, *21*, 136.

[18] R. L. Motard, A. W. Westerberg, *AIChE J*. **1981**, *27*, 725.

[19] J. F. Boston, S. L. Sullivan Jr, *The Can. J. Chem. Eng*. **1974**, *52*, 52 .

[20] R. B. Saeger, P. R. Bishnoi, *The Can. J. Chem. Eng*. **1986**, *64*, 759.

[21] J. Jelínek, *Comput. & Chem. Eng*. **1988**, *12*, 195 .

[22] J. H. Wegstein, *Commun. ACM* **1958**, *1*, 9 .

[23] O. Orbach, C. M. Crowe, *The Can. J. Chem. Eng*. **1971**, *49*, 509 .

[24] C. G. Broyden, *Math. computation* **1965**, *19*, 577 .

[25] N. J. van Eck, L. Waltman, *Sci*. **2010**, *84*, 523.

[26] Clarivate Analytics, Web of Science Core Collection **2020**, accessed on 23 April 2020. URL http://apps.webofknowledge.com

[27] W.-J. Hung, I.-K. Lai, Y.-W. Chen, S.-B. Hung, H.-P. Huang, M.-J. Lee, C.-C. Yu, *Ind. & Eng. Chem. Res*. **2006**, *45*, 1722.

[28] C. Pirola, F. Galli, F. Manenti, M. Corbetta, C. L. Bianchi, *Ind. & Eng. Chem. Res*. **2014**, *53*, 18063.

[29] Y. Xiao, W. Cai, H. Sun, F. Shi, G. Li, *The Can. J. Chem. Eng*. **2018**, *96*, 722.

[30] S. Wu, K. Guo, C. Liu, W. Qi, T. Zhang, H. Liu, *The Can. J. Chem. Eng*. **2018**, *96*, 2431.

[31] C.-K. Kuei, M.-D. Lee, *The Can. J. Chem. Eng*. **1991**, *69*, 347.

[32] N. A. Bahari, W. N. R. Wan Isahak, M. S. Masdar, Z. Yaakob, *Int. J. Energy Res*. **2019**, *43*, 5128.

[33] D. Previtali, M. Longhi, F. Galli, A. D. Michele], F. Manenti, M. Signoretto, F. Menegazzo, C. Pirola, *Fuel* **2020**, *274*, 117804.

[34] B. Duhoux, R. T. Symonds, R. Hughes, P. Mehrani, E. J. Anthony, A. Macchi, *The Can. J. Chem. Eng*. **2020**, *98*, 75.

[35] D. W. Keith, G. Holmes, D. S. Angelo], K. Heidel, *Joule* **2018**, *2*, 1573.

[36] J. C. Goncalves, A. E. Rodrigues, *The Can. J. Chem. Eng*. **2015**, *93*, 2205.

[37] P. Lutze, R. Gani, J. M. Woodley, *Chem. Eng. Process. Process. Intensif*. **2010**, *49*, 547 .

[38] E. Barbera, F. Mantoan, A. Bertucco, F. Bezzo, *The Can. J. Chem. Eng*. **2020**, *n/a*.

[39] K. D. Dahm, R. P. Hesketh, M. J. Savelski, *Chem. Eng. Educ*. **2002**, *36*, 192.

[40] A. M. Borreguero, J. L. Valverde, J. M. García-Vargas, L. Sánchez-Silva, *Comput. Appl. Eng. Educ*. **2019**, *27*, 1267.

[41] S. Yang, L. Yang, C. He, *Process. Saf. Environ. Prot*. **2001**, *79*, 329 .

[42] C. Pirola, *Educ. for Chem. Eng*. **2019**, *28*, 54 .

[43] T. M. Komulainen, R. Enemark-Rasmussen, G. Sin, J. P. Fletcher, D. Cameron, *Educ. for Chem. Eng*. **2012**, *7*, e153 .

For Peer Review

[44] L. Calvo, C. Prieto, *Educ. for Chem. Eng*. **2016**, *17*, 65 .

[45] A. Rafael, F. Bernardo, L. Ferreira, M. Rasteiro, J. Teixeira, *Educ. for Chem. Eng*. **2007**, *2*, 20 .

[46] L. Fregolente, H. Venturelli, J. Rodrigues, E. Da Silva, I. Diniz, R. Wolf Maciel Maria, *Chem. Eng. Transactions* **2018**, *69*, 427.

[47] S. Yang, L. Yang, C. He, *Process. Saf. Environ. Prot.* **2001**, *79*, 329 .

[48] R. M. Ghoniem, H. Abas, H. Bdair, *Procedia Comput. Sci*. **2018**, *135*, 490 .

[49] C. Foreman, M. Hilditch, N. Rockliff, H. Clarke, A Comparison of Student Perceptions of Physical and Virtual Engineering Laboratory Classes, Springer International Publishing, Cham **2020**, pp. 151–167.

[50] C. Pirola, F. Galli, C. Peretti, *Comput. & Chem. Eng. J.* **2020**, accpted.

[51] P. R. C. Camargo, E. de Barros Basso, FCC Training simulator–Petrobras experience **2005**.

[52] J. Isimite, F. Baganz, V. C. Hass, *J. Chem. Technol. & Biotechnol*. **2018**, *93*, 1529.

[53] D. S. Patle, D. Manca, S. Nazir, S. Sharma, *Virtual Real.* **2019**, *23*, 293.

[54] J. Haydary, Reactors, chapter 5, John Wiley & Sons, Ltd **2019**, pp. 101–124.

[55] S. S. Elnashaie, Modelling, simulation and optimization of industrial fixed bed catalytic reactors, volume 7, CRC Press **1994**.

[56] J. Haydary, Chemical Process Design and Simulation: Aspen Plus and Aspen Hysys Applications, John Wiley & Sons **2019**.

[57] M. A. Mirzakhani, N. Tahouni, M. H. Panjeshahi, *Energy* **2017**, *130*, 382.

[58] M. Sardarmehni, N. Tahouni, M. H. Panjeshahi, *Energy* **2017**, *127*, 623.

[59] M. V. Kanischev, L. M. Ulyev, R. E. Chibisov, M. A. Vasilyev, *Chem. Eng. Transactions* **2018**, *70*, 1099.

[60] Q. Zhang, M. Yang, G. Liu, X. Feng, *J. Clean. Prod.* **2016**, *112*, 4799.

[61] S. S. Chauhan, S. Khanam, *Energy* **2019**, *173*, 364.

[62] R. R. Tan, D. C. Y. Foo, K. B. Aviso, D. K. S. Ng, *Appl. Energy* **2009**, *86*, 605.

[63] G. Venkatesh, *Resour. Environ. Inf. Eng*. **2019**, *1*, 1.

[64] D. I. Garnett, G. S. Patience, *Chem. Eng. Prog*. **1993**, *89*, 76.

[65] R. Williams Jr., *Chem. Eng*. **1947**, *54*, 124 .

[66] G. S. Patience, D. C. Boffito, *J. Adv. Manuf. Process.* **2020**, *2*, e10039.

[67] H. J. Lang, *Chem. Eng* **1947**, *54*, 27.

[68] H. J. Lang, *Chem. Eng*. **1947**, *54*, 130 .

[69] H. J. Lang, *Chem. Eng*. **1948**, *55*, 112 .

[70] K. M. Guthrie, *Chem. Eng*. **1969**, *76*, 114.

[71] K. Guthrie, Process Plant Estimating, Evaluation, and Control, Craftsman Book Company of America **1974**.

[72] G. D. Ulrich, P. T. Vasudevan, Chemical Engineering Process Design and Economics: A Practical Guide, Process Publishing, Duhram, New Hampshire **2007**.

[73] G. D. Ulrich, P. T. Vasudevan, *Chem. Eng*. **2006**, 66 . URL http://terpconnect.umd.edu/~NSW/chbe446/HowToEstimateUtilityCosts-UlrichVasudevan2006.pdf

[74] R. A. Turton, J. A. Shaeiwitz, D. Bhattacharyya, W. B. Whiting, Analysis, synthesis and design of chemical processes, 5th edition, Pearson Education **2018**.

[75] Y. Feng, G. P. Rangaiah, *Chem. Eng*. **2011**, 22 . URL http://sgpwe.izt.uam.mx/files/users/uami/rmro/2122096/S11/Evaluating_capital_cost_estimation_program.pdf

[76] W. D. Seider, J. D. Seader, D. R. Lewin, Product & Process Design Principles: Synthesis, Analysis and Evaluation, (With CD), John Wiley & Sons **2009**.

[77] R. Sinnott, G. Towler, Chemical engineering design: SI Edition, Butterworth-Heinemann **2019**.

[78] N. Patel, N. Padhiyar, *Chem. Eng. Res. Des.* **2017**, *119*, 160 .

[79] Practical optimization: a gentle introduction **2006**. URL https://eclass.gunet.gr/modules/document/file.php/LABGU361/PRACTICAL%20OPTIMIZATION%20GENTLE%20INTRODUCTION%202015.pdf

[80] L. T. Biegler, Nonlinear programming: concepts, algorithms, and applications to chemical processes, volume 10, Siam **2010**.

For Peer Review

[81] R. A. Stubbs, S. Mehrotra, *Math. Program.* **1999**, *86*, 515 .

[82] J. A. Nelder, R. Mead, *The Comput. J.* **1965**, *7*, 308.

[83] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, et al., Evolutionary algorithms for solving multi-objective problems, volume 5, Springer **2007**.

[84] J. M. Ponce-Ortega, L. G. Hernandez-Perez, Optimization of Process Flowsheets through Metaheuristic Techniques, Springer **2019**.

[85] F. Galli, C. Pirola, D. Previtali, F. Manenti, C. L. Bianchi, *J. Clean. Prod.* **2018**, *171*, 147.

[86] J. J. Quiroz-Ramirez, E. Sanchez-Ramirez, S. Hernandez, J. H. Ramirez-Prado, J. G. Segovia-Hernandez, *Ind. & Eng. Chem. Res.* **2017**, *56*, 1823.

[87] J. J. Quiroz-Ramirez, E. Sanchez-Ramirez, S. Hernandez-Castro, J. G. Segovia-Hernandez, J. M. Ponce-Ortega, *ACS Sustain. Chem. & Eng.* **2017**, *5*, 4018.

[88] J. C. Eslick, D. C. Miller, *Comput. & Chem. Eng.* **2011**, *35*, 1488 .

[89] S. Sharma, G. P. Rangaiah, *Comput. & chemical engineering* **2013**, *56*, 155.

[90] B. Linnhoff, J. R. Flower, *AIChE J.* **1978**, *24*, 633.

[91] I. C. Kemp, Pinch analysis and process integration: a user guide on process integration for the efficient use of energy, Elsevier **2011**.

[92] B. Linhoff, J. Flower, *AIChE J.* **1978**, *24*, 642.

[93] K. Brown, Process Integration Initiative: A Review of the Process Integration Activities Funded Under the Energy Efficiency R and D Programme, Energy Technology Support Unit (ETSU) **1989**.

[94] T. Umeda, J. Itoh, K. Shiroko, *Chem. Eng. Prog.* **1978**, *74*, 70.

[95] P. M. Pereira, M. C. Fernandes, H. A. Matos, C. P. Nunes, *Appl. Therm. Eng.* **2017**, *114*, 523.

[96] E. Rosen, ACS Publications **1980**.

[97] N. M. Kazerooni, R. Eslamloueyan, M. Biglarkhani, *Int. J. Ind. Chem.* **2019**, *10*, 107.

[98] Y. Xu, H. Dinh, Q. Xu, F. T. Eljack, M. M. El-Halwagi, *J. Clean. Prod.* **2020**, 120129.

[99] P. Thomas, S. Bhattacharyya, A. Patra, G. Rao, *Comput. & chemical engineering* **1998**, *22*, 1515.

[100] M. Saidi, H. Kadkhodayan, *J. Environ. Chem. Eng.* **2020**, *8*, 103772.

[101] H. Jafari, A. Sheikhi, R. Sotudeh-Gharebagh, *The Can. J. Chem. Eng.* **2020**, *98*, 1003 .

[102] Y. Shen, Z. Tang, D. Guo, C. Wu, D. Wang, A. Jiang, Y. Wen, *Appl. Therm. Eng.* **2019**, *160*, 113983.

[103] NIST Standard Reference Database Number 69 **2020**. URL https://webbook.nist.gov/chemistry/

[104] Engineering Data and Technical References to Solve Engineering Problems **2020**. URL https://app.knovel.com/web/index.v?jsp=main

[105] B. E. Poling, J. M. Prausnitz, J. P. O'Connell, The properties of gases and liquids, 5th edition, Mcgraw-hill New York **2001**.

[106] T. Jensen, A. Fredenslund, P. Rasmussen, *Ind. & Eng. Chem. Foundamentals* **1981**, *20*, 239.

[107] W. Wakeham, M. Assael, *BULGARIAN CHEMICAL COMMUNICATIONS* **2019**, *51*, 9 .

[108] R. Gani, J. P. O'Connell, *Comput. & Chem. Eng.* **1989**, *13*, 397 .

[109] E. C. Carlson, *Chem. Eng. Prog.* **1996**, 35 .

[110] J. G. Hayden, J. P. O'Connell, *Ind. & Eng. Chem. Process. Des. Dev.* **1975**, *14*, 209.

[111] J. Gmehling, M. Kleiber, B. Kolbe, J. Rarey, Chemical thermodynamics for process simulation, John Wiley & Sons **2019**.

[112] K. C. Chao, J. D. Seader, *AIChE J.* **1961**, *7*, 598.

[113] B. I. Lee, M. G. Kesler, *AIChE J.* **1975**, *21*, 510.

[114] O. Redlich, J. N. S. Kwong, *Chem. Rev.* **1949**, *44*, 233.

[115] D.-Y. Peng, D. B. Robinson, *Ind. & Eng. Chem. Fundamentals* **1976**, *15*, 59.

[116] P. A. Gupte, P. Rasmussen, A. Fredenslund, *Ind. & Eng. Chem. Fundamentals* **1986**, *25*, 636.

[117] G. M. Wilson, *J. Am. Chem. Soc.* **1964**, *86*, 127.

[118] H. Renon, J. M. Prausnitz, *AIChE J.* **1968**, *14*, 135.

[119] D. S. Abrams, J. M. Prausnitz, *AIChE J.* **1975**, *21*, 116.

For Peer Review

[120] A. Fredenslund, R. L. Jones, J. M. Prausnitz, *AIChE J.* **1975**, *21*, 1086.

[121] S. T. Holland, R. Abbas, D. Hildebrandt, D. Glasser, *Ind. & engineering chemistry research* **2010**, *49*, 327.

[122] R. C. Pattison, M. Baldea, *AIChE J.* **2014**, *60*, 4104.

[123] Chemistry Effects Predicting Simulation Software **2020**.
URL https://www.ansys.com/products/fluids/ansys-chemkin-pro

[124] D. F. Rivas, D. C. Boffito, J. A. F. Albanese, J. Glassey, N. Afraz, H. Akse, K. Boodhoo, R. Bos, J. Cantin, Y. W. Chiang, J. M. Commenge, J. L. Dubois, F. Galli, J. Harmsen, S. Kalra, F. Keil, R. Morales-Mendez, F. J. Navarro-Brull, T. Noel, G. Patience, D. Reay, R. Santos, A. Smith-Schoettker, A. Stankiewicz, H. van den Berg, T. V. Gerven, J. van Gestel, M. van der Stelt, M. van de Ven, R. Weber, *Educ. for Chem. Eng.* **2020**, *32*, 1 .

[125] D. F. Rivas, K. Boodhoo, R. Morales-Mendez, Y. W. Chiang, F. Navarro, T. V. Gerven, J. van Gestel, R. Weber, S. Kalra, D. C. Boffito, D. Reay, J. A. F. Albanese, F. Keil, T. Noel, J. Glassey, A. Stankiewicz, H. van den Berg, F. Galli, K. Ogden, R. Santos, A. Smith-Schoettker, J. Cantin, H. Akse, G. Patience, J. M. Commenge, J. Harmsen, J. L. Dubois, N. Afraz, R. Bos, *Educ. for Chem. Eng.* **2020**, *32*, 15 .

[126] CHEMSAFE **2020**.
URL https://dechema.de/en/chemsafe.html

[127] A. Kummer, T. Varga, *J. Loss Prev. Process. Ind.* **2019**, *58*, 22 .

[128] J. E. Coon, M. Kush, M. C. Rowland, J. R. Cunningham, Presented at the AIChE Spring Meeting, Citeseer **1998**, pp. 1 – 11.

For Peer Review

Received ; | Revised: 00 Month 0000 | Accepted: 00 Month 0000

DOI: xxx/xxxx

MINI-REVIEW

Experimental Methods in Chemical Engineering: Process Simulation

Jacopo De Tommaso[1] | Francesco Rossi[2] | Nooshin Moradi[3] | Carlo Pirola[4] | Gregory S. Patience[1] | Federico Galli*[1]

[1]Chemical Engineering, Polytechnique Montréal, C.P. 6079, Succ. “CV”, Montréal, H3C 3A7 Québec, Canada
[2]School of Chemical Engineering, Pordue University, Davidson School of Chemical Engineering, Forney Hall of Chemical Engineering, 480 Stadium Mall drive, West Lafayette, 47907-2100 IN, USA
[3]School of Chemical and Petroleum Engineering, Shiraz University, 71348-51154, Shiraz Iran
[4]Dipartimento di Chimica, Universitá degli Studi di Milano, via Golgi 19, 20133, Milano, Italy

Correspondence
*Corresponding author: Federico Galli, Polytechnique Montréal. Email: federico.galli@polymtl.ca

Summary

Process simulation software designs equipment, simulates operations, optimizes a plant’s configuration (heat exchangers network, for example), estimates operating and capital expenses, and serves as educational tools. However, mastering the theoretical background minimizes common mistakes such as applying an incorrect thermodynamic method, selecting improper algorithms in the case of tear systems, and set irrational system specifications. Engineers and researchers will exploit this tool more often in the future as constant advancements in simulation science as well as new models are released continually. Process simulators ease the building of digital twins and thus will facilitate implementation of the industry 4.0 guidelines. We highlight the mathematical and technical features of process simulators, as well as the capabilities and the fields of applications. A bibliometric map of keywords from articles citing Aspen+, Aspen plus, Hysys, and Pro/II indexed by *Web of Science* between 2017 and 2020 identified the main research clusters such as: Design; Optimization; Energy or Exergy; Biomass; H_2 and CO_2 capture; Thermodynamics; separations and Techno-Economic analysis.

KEYWORDS:
Process simulation, Design, Thermodynamics, Optimization, Cost estimation

1 | INTRODUCTION

Process simulators solve mathematical equations not only to characterize reactors, distillation columns, heat exchangers, pumps, compressors and other unit operations but also for process design and optimization. Furthermore, they offer engineers a tool to estimate plant costs—both operating expenses (OPEX) and capital expense (CAPEX). Users input operating pressure, temperature, flow rates, composition, and thermodynamic packages and the software computes mass and energy balances for each stream and module. Examples of the first software date back to 1960s with PACER developed by McMaster University for educational purposes. The University of Houston developed in collaboration with industry CHESS in 1968. It simulated the phase equilibria of more than 70 hydrocarbons.[1] PROCESS, by Simulation science, was distributed in 1966 and simulated distillation columns. It became then PRO/II, first distributed by Scheider-Electrics and then by AVEVA. Fortran expanded the capabilities of these programs in the 70s. In the late 70s, MIT developed Advanced System for Process ENgineering (ASPEN) that AspenTech™ commercializes in the 80s. Concurrently Profs. Bishnoi and Svercek at the University of Calgary developed HYSYS. In the late 1980s and early 1990s the PC-based simulators were developed.[2] In 2002 AspenTech™ acquired Hyprotech; however, the Federal Trade Commission required AspenTech to divest the HYSYS because they deemed it contravened anti-trust laws. Honeywell became the owner of HYSYS and created UniSim. Eventually, AspenTech™ re-acquired HYSYS in 2016 and now 91 simulators are on the market including open source software.

This mini-review on process simulation is part of a series of articles dedicated to experimental methods in chemical engineering.[3] Although practicing engineers and students use these simulators for plant design and optimization, researchers also apply them to calculate thermodynamics, identify equilibrium compositions of reactions, verify the feasibility of separation operations, and thus optimize experimental designs. Here we discuss these features but address the mathematical approaches to solve these large systems of equations, resolution algorithms, and applications. We describe the main sources of error and how to avoid common mistakes that arise at the set up of the flowsheet and include a bibliometric survey that highlights the major applications.

For Peer Review

2 | THEORY

Steady-state models of most unit operations are nonlinear systems of either algebraic or differential-algebraic equations (e.g., PFR reactors and pipes). Discretization techniques convert differential -algebraic equations into a system of algebraic equations, so any steady-state process flowsheet is equivalent to a large-scale system of nonlinear equations. Therefore, in this section of the manuscript, we first review the rationale of methods that solve these systems of nonlinear equations, and then explain how simulation packages use them to solve process flowsheets. We discuss both modular and equation-oriented strategies for flowsheet simulation, comment on their advantages and disadvantages, and provide guidelines on when/how to apply each approach.[4, 5]

2.1 | Numerical methods for systems of nonlinear equations

The most general mathematical formulation of a system of nonlinear equations, Eq 1, where every function g_i represents a nonlinear expression of the unknown variables x_i (in a process flowsheet, the functions g_i represent steady-state mass, energy and momentum balances, and discretization thereof or design specifications, whereas the quantities x_i are process variables, e.g. temperatures, pressures, molar/mass fractions and flowrates).

$$\begin{cases} g_1(x_1, x_2, x_3, ..., x_N) = 0 \\ g_2(x_1, x_2, x_3, ..., x_N) = 0 \\ \dots \\ g_N(x_1, x_2, x_3, ..., x_N) = 0 \end{cases} \Leftrightarrow g(\boldsymbol{x}) = 0 \qquad (1)$$

The objective to solve this nonlinear system ($\boldsymbol{x_s}$), which corresponds to a root of all the functions g_i ($g_i(\boldsymbol{x_s}) = 0$). There exist no analytical method to accomplish this task, so we only estimate an approximate value of $\boldsymbol{x_s}$ through numerical algorithms—Newton type and fixed-point methods. Both these iterative procedures rely on a user-supplied initial value, $\boldsymbol{x_0}$ and iteration $(\boldsymbol{x_n})_{n\geq 1}$ until they approach the solution vector $\boldsymbol{x_s}$. These two families of algorithms differ with respect to the procedure they apply to compute $(\boldsymbol{x_n})_{n\geq 1}$, their efficiency and robustness, and their application domain.

Starting from the last known iteration, $\boldsymbol{x_n}$, Newton-type methods calculate $\boldsymbol{x_{n+1}}$ with the following procedure:

1. Exact or approximate linearization of the nonlinear systems at $\boldsymbol{x_n}$. This step requires calculation of the Jacobian matrix at $\boldsymbol{x_n}$ or of an estimate thereof (the matrices $\boldsymbol{J}(\boldsymbol{x_n})$ and B_n, respectively);

2. Solution of the linearized system of equations and calculation of the search direction $\boldsymbol{d_n}$, according to the first line of Eq. 2 and Eq. 3;

3. Computation of the optimal step length $\boldsymbol{a_n}$ through a line search strategy, performed along $\boldsymbol{d_n}$;[6]

4. Calculation of $\boldsymbol{x_{n+1}}$ from $\boldsymbol{x_n}$, $\boldsymbol{a_n}$, and $\boldsymbol{d_n}$, according to the second line of Eq. 2 and Eq. 3.

$$\begin{cases} \boldsymbol{J}(\boldsymbol{x_n})\boldsymbol{d_n} = -g(\boldsymbol{x_n}) \\ \boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{a_n}\boldsymbol{d_n} \end{cases} \wedge \boldsymbol{J}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_N} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_N}{\partial x_1} & \frac{\partial g_N}{\partial x_2} & \cdots & \frac{\partial g_N}{\partial x_N} \end{pmatrix} \qquad (2)$$

$$\begin{cases} B_n\boldsymbol{d_n} = -g(\boldsymbol{x_n}) \\ \boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{a_n}\boldsymbol{d_n} \end{cases} \wedge B(n) \approx \boldsymbol{J}(\boldsymbol{x_n}) \qquad (3)$$

These steps are repeated until convergence criteria are met, e.g. the magnitude of the search direction ($\|\boldsymbol{d_n}\|$) becomes sufficiently small, or the sequence $(\boldsymbol{x_n})_{n\geq 1}$ diverges (Fig. 1). The sequence $(\boldsymbol{x_n})_{n\geq 1}$ follows the same type of pattern (Fig. 1) for systems of any number of equations.

Newton-type methods do not guarantee convergence to a solution of the nonlinear system, starting from any random initial guess as this family of algorithms is only locally convergent. However, their rate of convergence is superlinear. Therefore, they are very effective when a good initial guess is available. Another feature of Newton-type methods is their capability of solving nonlinear systems of *arbitrarily* large scale, provided that adequate computing power is available.

Fixed-point methods require a reformulation of the original system of nonlinear equations, which consists of re-writing

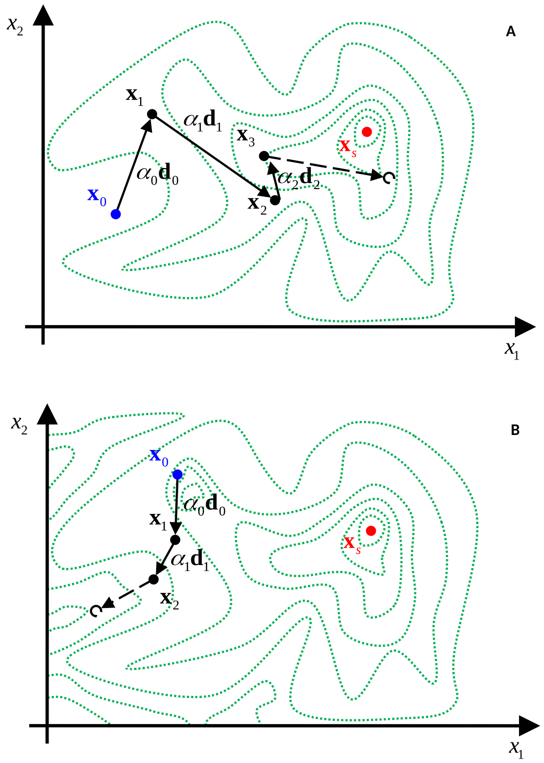

FIGURE 1 Typical sequence of iterations generated by a Newton-type method. (A - convergent sequence; B - divergent sequence). From the initial guess $\boldsymbol{x}_0$, the algorithm calculates the direction vector $\boldsymbol{d}_i$ and the step length $\boldsymbol{a}_i$, moving towards $\boldsymbol{x}_S$ or without converging.

every equation $g_i(\boldsymbol{x}) = 0$ in the same form as Eq. 4 (this reformulation is always possible, although not necessarily unique, and may affect the convergence properties of the algorithm).

$$x_i = f_i(\boldsymbol{x}) \quad \forall i \in [1, N] \tag{4}$$

After this preliminary step, starting from the last known iteration $\boldsymbol{x}_n$, the next iteration $\boldsymbol{x}_{n+1}$ is calculated through the following procedure:

1. Calculation of the functions f_i at $\boldsymbol{x}_n$;
2. Selection of values for all the relaxation factors $\boldsymbol{a}_{i,n}$;
3. Application of Eq. 5.

$$\boldsymbol{x}_{i,n+1} = \boldsymbol{a}_{i,n} f_i(\boldsymbol{x}_n) + (1 - \boldsymbol{a}_{i,n})\boldsymbol{x}_{i,n} \quad \forall i \in [1, N] \tag{5}$$

These steps are repeated until convergence criteria are met, e.g. the norm of the difference between two consecutive iterations ($\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\|$) becomes sufficiently small, or the sequence $((\boldsymbol{x}_n)_{n\geq1}$ diverges (the direct substitution recursion matches Eq. 5 with all the relaxation factors $\boldsymbol{a}_{i,n} = 1$). The sequence $(\boldsymbol{x}_n)_{n\geq1}$ follows a similar pattern for systems of two or more equations (Fig. 2).

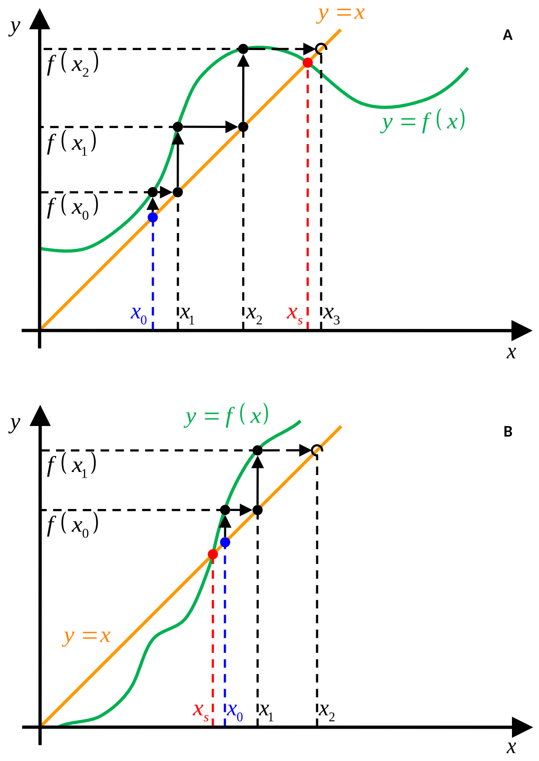

FIGURE 2 Typical sequence of iterations generated by direct substitution (A–convergent sequence; B–divergent sequence). From the initial guess $\boldsymbol{x}_0$, the algorithm calculates $\boldsymbol{x}_i$ and $f(\boldsymbol{x}_i)$, moving towards $\boldsymbol{x}_S$ or without converging.

As with Newton-type methods, fixed-point schemes do not guarantee convergence to a solution of the nonlinear system, starting from any random initial guess (these algorithms are only locally convergent), but they are usually less sensitive to the initial value. On the other hand, their rate of convergence is slower than Newton-type methods. Thus, they are particularly suitable for situations in which no good initial guess can be computed. Note that, as opposed to Newton-type algorithms, fixed-point methods can only solve small-scale/medium-scale nonlinear systems, regardless of the amount of computing power available. This limitation considerably restricts their application domain.

2.2 | Computational Strategy

The barebones of a flowsheeting software differs from those of others based on the computational strategies. The main techniques are:

- Sequential Modular Approach (**SM**)
- Equation Oriented Approach (**EO**)
- Simultaneous Modular Approach

The SM solves the process units in sequence, starting from the feed and tear the common streams in case of recycle. Where an inlet stream is given, each block computes its outlet stream. However, in the presence of a recycle, an iteration is required. Very sturdy and reliable, this is the default method of commercial and general software, especially for steady-state calculations. On the other hand, The SM is weak and time-consuming for[7]:

- Highly recycled processes;
- Highly-Heat integrated processes;
- Optimization;
- Simultaneous flowsheet and design specification loops; and,
- Dynamic simulations.

In the EO approach the whole process is solved simultaneously as a system of nonlinear algebraic equation. EO requires a more experienced simulator, is difficult to debug, and works well only when the initial value is close to the solution. Nevertheless, it works better where SM is weak. The Simultaneous Modular Approach is a combination of both SM and EO.[8] The flowsheet is solved in an EO fashion, while the singular unit are solved sequentially.[9] So far, this approach finds application in academia,[10–12] but not in commercial software.

2.3 | Modular solution (SM) strategies for process flowsheets

Any chemical process is a set of unit operations, primarily connected in series as the number of recycles is usually significantly smaller than the total number of connections. Modular solution strategies for process simulation leverage this sequential nature of chemical processes to first simplify and then perform flowsheet calculations. The rationale of these solution strategies is:

1. Given a certain process flow diagram (PFD), we first convert it into a directed graph (the simulation flow diagram (SFD)), whose vertices and edges represent units and streams, respectively (Fig. 3).
2. Then, well-established algorithms, e.g. the connection matrix method[13]or the depth-first search and backtrack method[14], detect all the simple cycles in the flowsheet graph (Fig. 3), which corresponds to identifying all the recycles in the original process flowsheet.
3. Subsequently, we break all the simple cycles by tearing a number of edges in the flowsheet graph (Fig. 4), thus transforming the latter into a directed tree (the flowsheet tree). This graphical operation corresponds to breaking all the recycles in the original process flowsheet, by replacing specific process streams (the tear streams) with pairs of new streams (the artificial streams), so as to generate a simplified process flowsheet, in which all the unit operations can be solved in series, once half the artificial streams have been assigned (the remaining artificial streams become simulation outputs). Note that every recycle stream is literally torn at least once, thus the name *tear stream*.
4. Finally, we exploit the connectivity information, stored in the flowsheet tree, to solve all the unit operations and compute all the process streams in the simplified process flowsheet, under the additional requirement that every pair of artificial streams be identical (these artificial constraints ensure that all the recycles are enforced). This is mathematically equivalent to solving Eq. 6 (the recycle problem), in which the vector $\boldsymbol{x}$ represents the process variables of the artificial streams that must be assigned, and the function $f(\boldsymbol{x})$ denotes the process variables of the artificial streams that are simulation outputs. $f(\boldsymbol{x})$ is an implicit function of $\boldsymbol{x}$, as we can only evaluate it by solving some/all of the unit operations in the simplified process flowsheet).

$$\boldsymbol{x} - f(\boldsymbol{x}) = 0 \qquad (6)$$

Eq. 6 contains only a small fraction of the nonlinear equations, which describe the original process flowsheet of interest, provided that the latter contains a small number of recycles (the scale of Eq. 6 is indeed proportional to the number of recycles). Therefore, modular solution strategies are suitable for simulation of process flowsheets of *arbitrary* scale, which contain few recycles. On the other hand, they should never be used to solve process flowsheets with many recycles, as the numerical solution of Eq. 6 becomes impractical.

2.3.1 | Selection of the optimal tear streams

The identification of optimal tear streams is an element of any modular solution strategy for process flowsheets because the number and features of the tear streams influence the size and numerical properties of the recycle problem (Eq. 6). Therefore,

For Peer Review

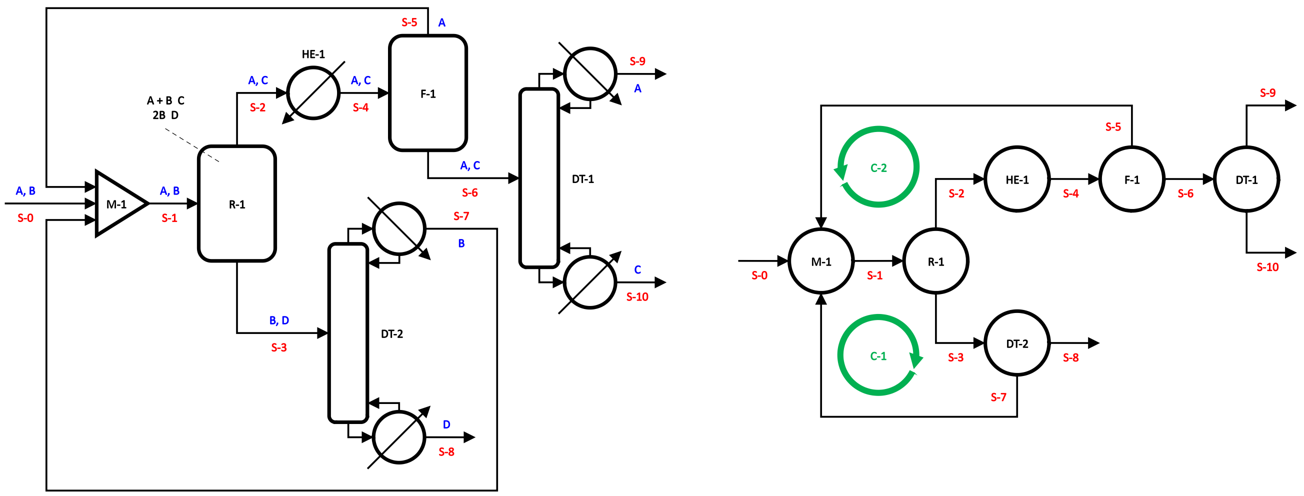

FIGURE 3 Simple process flowsheet complemented with its graph representation (S-N – Nth process stream; C-N – Nth simple cycle or Nth recycle).

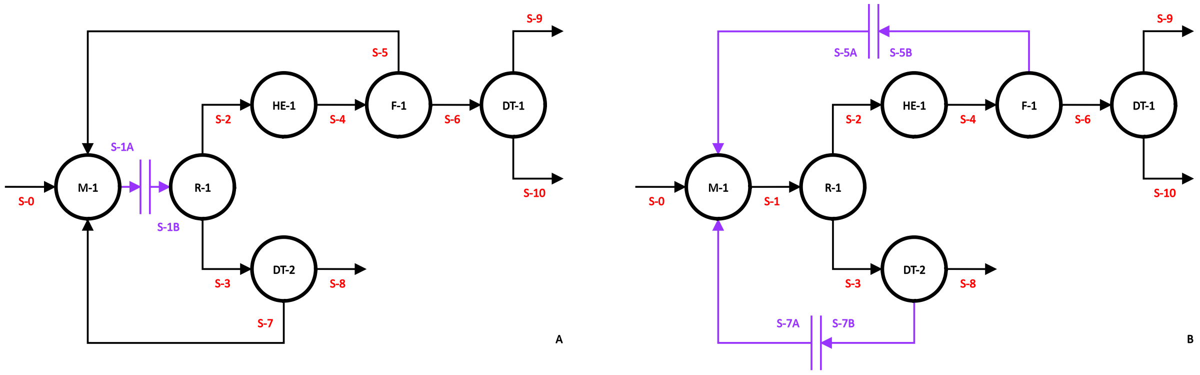

FIGURE 4 Reduction of the flowsheet graph to a directed tree by elimination of all the simple cycles (A - optimal tear streams according to Barkley and Motard[15] and Christensen and Rudd;[16] B - optimal tear streams according to Upadhye and Grens[17] and Motard and Westerberg.[18]).

many different tearing algorithms have been proposed over the years. However, all of these algorithms can be grouped in three different families, which rely on as many different underlying ideas:

- According to Barkley and Motard,[15] the tear streams should be selected as the smallest set of process streams that break all the recycles in the process flowsheet (Fig. 4).
- According to Christensen and Rudd,[16] the tear streams should be selected as the set of process streams that break all the recycles in the process flowsheet and minimize the size of the recycle problem (Fig. 4).
- According to Upadhye and Grens[17] and Motard and Westerberg,[18] the tear streams should be selected as the set of process streams that break all the recycles in the process flowsheet, under the additional requirement that the number of times every recycle is torn be minimum (Fig. 4).

Although none of these three alternatives is fully satisfactory, the best option depends on the numerical method, used to solve the recycle problem. More specifically, if we solve the recycle problem with Newton-type methods, then the second option represents the most appropriate choice from a computational perspective. On the other hand, if we solve the recycle problem with fixed-point algorithms, then the third option is the most suitable from a computational point of view (this tearing scheme maximizes the rate of convergence of fixed-point methods, when they are applied to Eq. 6). For computational (and historical) reasons, all the main software were originally designed as SM simulators. Hence, the first developers focused on the implementation of a single tearing algorithm that worked best in a SM environment. Therefore, most process simulation packages implement a single tearing algorithm, which often resembles the third, so the user's choices are limited to what is available in existing process simulation software. Due to the mathematical complexity of the tearing algorithm, we recommend a non expert user to operate the default method.

For Peer Review

2.3.2 | Solution of the single unit operation models

A key step of all modular solution strategies for process simulation is the solution of the recycle problem (Eq. 6), which in turn requires repeated calculations of single unit operations. Since models of different types of unit operations exhibit unique features, dedicated numerical methods to solve each and every one of them have been developed over the years. However, all these numerical algorithms belong to either partitioning methods or simultaneous methods. Partitioning methods decompose unit operation models into blocks of equations, which are then solved iteratively, in a predefined sequence, until convergence criteria are met or the iterations diverge. Internally, they rely on fixed-point methods, thus exhibit the same numerical properties as this family of algorithms. Simultaneous methods do not decompose unit operation models but rather solve them as nonlinear systems. Since they apply Newton-type methods (possibly complemented with some parametric continuation approach), they exhibit similar numerical properties as these types of algorithms. Most process simulation packages implement simultaneous methods for unit operations but offer both partitioning and simultaneous alternatives for specific types of unit operations, e.g. distillation, absorption and stripping towers (for completeness, the most common partitioning solution scheme for separation towers is the inside-out method by Boston and Sullivan,[19] Saeger and Bishnoi[20] and Jelinek[21]). As a rule of thumb, we recommend partitioning methods for solving mildly non-ideal separations and simultaneous methods for solving highly non-ideal or reactive separations. In addition, simultaneous methods may be more appropriate for specific tasks, such as running case studies and/or performing flowsheet optimization, because of their higher computational efficiency. Additional recommendations on which numerical solution scheme to choose for specific simulation problems are usually available on the user guides of process simulation packages.

2.3.3 | Solution of the recycle problem

The last key component of any modular solution strategy for process flowsheets is the solution of the recycle problem (Eq. 6), which apply either fixed-point or Newton-type methods. In the 60s and 70s, we witnessed the birth of several new numerical algorithms, specifically designed to solve this problem. Nowadays, those that are still implemented in process simulation packages are the direct substitution method, the Wegstein method,[22] the dominant eigenvalue method,[23] the Newton-Raphson method and the Broyden method.[24] The rationale of each are:

- The direct substitution method is the simplest and oldest fixed-point algorithm, whose recursion matches Eq. 5 with all the relaxation factors $\boldsymbol{a}_{i,n} = 1$.
- The Wegstein method is an accelerated version of direct substitution, in which the relaxation factors $\boldsymbol{a}_{i,n}$ are computed by Eq. 7. We emphasize that Eq. 7 relies on the assumption that every variable of the recycle problem is independent of the others, so this numerical method may not perform satisfactorily when the Jacobian matrix of $f(\boldsymbol{x})$ (Eq. 6) is not diagonally dominant.
- The dominant eigenvalue method is an improved version of Wegstein, in which the relaxation factors $\boldsymbol{a}_{i,n}$ are computed by Eq 8. In this case, Eq 8 relies on the assumption that the Jacobian matrix of $f(\boldsymbol{x})$ (Eq. 6) has a dominant eigenvalue, so this numerical method may not perform satisfactorily when several variables of the recycle problem are equally important.
- The Newton-Raphson method is the simplest and oldest Newton-type scheme, whose recursion matches Eq. 2. It relies on the actual Jacobian of Eq. 6, so it may be more computationally efficient than Broyden when the recycle problem is a small-scale/medium-scale nonlinear system.
- The Broyden method is a Newton-type scheme, whose recursion matches Eq. 3. It relies on an estimate of the Jacobian of Eq. 6, computed by Eq. 9, so it may be more computationally efficient than Newton-Raphson when the recycle problem is a large-scale nonlinear system.

$$\boldsymbol{a}_{i,n} = \frac{1}{1 - \boldsymbol{s}_{i,n}} \wedge \boldsymbol{s}_{i,n} = \frac{f_i(\boldsymbol{x}_n) - f_i(\boldsymbol{x}_{n-1})}{\boldsymbol{x}_{i,n} - \boldsymbol{x}_{i,n-1}} \quad \forall i \in [1, N] \quad (7)$$

$$\boldsymbol{a_{i,n}} = \frac{1}{1-\lambda_n^{MAX}} \wedge \lambda_n^{MAX} = \frac{\|\boldsymbol{x_n} - \boldsymbol{x_{n-1}}\|}{\|\boldsymbol{x_{n-1}} - \boldsymbol{x_{n-2}}\|} \forall i \in [1, N] \quad (8)$$

$$B_n = B_{n-1} + \frac{g(\boldsymbol{x_n}) - g(\boldsymbol{x_{n-1}}) - B_{n-1}\boldsymbol{d_{n-1}}}{\|\boldsymbol{d_{n-1}}\|^2}\boldsymbol{d}^T_{\boldsymbol{n-1}} \wedge B_0 = J(\boldsymbol{x_0}) \quad (9)$$

All of these numerical approaches exhibit advantages and limitations, which make them more adequate for specific types of process flowsheets. Tab. 1 collects a general set of recommendations to help select a numerical alternative for most real-world simulation problems.

2.4 | Equation-oriented (EO) solution strategies

As opposed to modular solution approaches, equation-oriented solution strategies for process simulation do not attempt to simplify flowsheet calculations by exploiting the sequential nature of chemical processes. They construct the large-scale sparse system of nonlinear equations, which describes the process flowsheet of interest and includes unit operation models, process streams and design specifications, and solve it using dedicated numerical methods. The numerical schemes, suitable for flowsheet simulation, are Newton-type methods, combined with some merit function minimization strategy and, very often, some parametric continuation approach, which helps the sequence of iterations converge whenever equations are highly nonlinear, their Jacobian matrix is locally rank deficient, and/or a good initial guess cannot be computed.[6]

Regardless of any specific numerical details, equation-oriented solution strategies solve any type of process flowsheets, independently of their scale and topology (note that the performance of these solution schemes is unaffected by the number of recycles in the process flowsheet, provided that adequate computing power is available). However, they should be primarily used to solve process flowsheets, which contain many recycles, as modular solution approaches are more computationally efficient for simulation of process flowsheets with few recycles.

3 | APPLICATIONS

The program VoSViewer generated a bibliometric map of keywords generated from articles indexed by Web of Science Core Collection (WoS) from 2017 to April 2020.[25, 26] We queried the database 4 times with ASPEN +, ASPEN Plus, HYSYS, and PROII as topics, while simultaneously excluding forest. (To avoid work related to Aspen trees and forests rather than process simulation). We combined the four databases into one and eliminated all duplicate and ended up with 2582 articles. The Web of Science category energy & fuels had 930 articles while the chemical engineering category was a close second with 824 articles followed by thermodynamics (314 articles), environmental engineering (224), and green & sustainable science and technology (215).

We retained about 90 of the keywords that were cited most frequently in these articles but excluded *simulation*, *performance*, *column*, *behaviour*, and *growth*. VOSViewer grouped the keywords into 5 clusters with Aspen+ at the centre of the red cluster with the most articles (433) and the cluster with the most keywords at 29 (Figure 5). This cluster includes topics related to biomass (193 articles)— ethanol and bioethanol (132), lignocellulosics, and cellulose (64). The two other major topics relate to catalysis/kinetics (162) and techno-economic analysis/economics (227) and other subjects related to biomass.

Process simulation discriminates between process alternatives (Design—246 articles), which comprises modelling (187 articles) and determines the dimensions of unit operations like separation (126) including distillation (137 articles) (reactive and extractive), heat integration (33), and dehydration (26). Hung et al., for example, demonstrated that recovering dilute acetic acid water solutions (between 30 % to 70 % by mass of acid) with reactive distillation, amyl alchols among C_1–C_5 alcohols minimizes the total annual cost (TAC), without any stream pretreatment.[27] Pirola et al., simulated an extractive distillation column employing p-xylene as entrainer to maximize acetic acid recovery.[28] Xiao et al. dimensioned a three-column process to transesterify ethylene glycol with methyl acetate, regressing reaction kinetics parameters and optimized the design minimizing the TAC.[29] Wu et al. designed a dividing-wall column, together with its control strategy, for reformed gasoline.[30] Biodiesel and esterification are concepts less related to process simulation as they are far from the map center. This is because first- and second-generation biofuels interest is dropping in recent years in favour of other green technologies for the conversion of green house gases, such as CO_2 into fuels.[31–33] In fact, the magenta cluster centered around H_2 (227 articles) concentrates on gasification (biomass+steam with 239 articles), syngas (141), and reactors and fluidized beds (84).

CO_2 capture (193 articles), together with CO_2 (162), and CH_4 dominate the yellow cluster that includes combustion (87), absorption (82), and Hysys (66). This cluster spans a large area covering technology related to carbon capture and sequestration (CCS). Duhoux et al., for example, calculated the optimal pressure and flowrate of a pressurized fluidized bed combustion that sequestrates carbon dioxide via Calcium looping.[34] *Joule* has published the most cited article since 2019 (101 citations) that describes a plant to capture 1 Mt CO_2 per year with aqueous KOH sorbent and a calcium caustic recovery loop.[35] In fact, the article spans many of the keywords in

For Peer Review

TABLE 1 Numerical methods for flowsheet simulation.

Numerical method	Classification	Robustness	Efficiency	Recommendations
Direct substitution	Fixed-point	Low	Low	Neither reliable nor efficient Avoid its use.
Wegstein	Fixed-point	Medium	Low	Suitable for flowsheets that contain weakly interacting recycles[a].
Dominant eigenvalue	Fixed-point	Medium	Low	Applicable to process flowsheets with weakly and/or strongly interacting recycles[a,b]. Use when Wegstein fails
Newton-Raphson	Newton-like	Low	High	Fast but sensitive to initial guesses. Useful for flowsheet optimization and/or running case studies.
Broyden	Newton-like	Low	Medium	Equivalent to Newton-Raphson, more efficient for large-scale process flowsheets.

[a]: A set of recycles exhibits weak interactions when changes in the properties of any recycle stream induce small variations in the properties of all the other recycle streams.

[b]: A set of recycles exhibits strong interactions when changes in the properties of any recycle stream cause consistent variations in the properties of all the other recycle streams.

For Peer Review

the map—unit operations, capital cost estimates, and heat and mass balances with ASPEN+.

The Optimization (266 articles) cluster is connected to both Design and CO_2 capture (Figure 5). Many of the subjects relate to Process Intensification, which includes energy (228 articles) systems (215), energy/efficiency (127), and exergy (163).[36, 37] Dynamic simulation analyzes process start-up and shut downs, and closely connects to process design.[38]

Process simulation is an educational tool to demonstrate many facets of design.[39, 40] Steady-state simulation represents the plant at stationary conditions while dynamic simulation is required for the plant control strategies, for start up and showdown operations, and for operator training by plant virtualization.[41] The suggested approach for undergraduate students is steady state simulation,[42] while dynamic simulation is appropriate for graduate level courses.[43] For instance, the Universidad Complutense of Madrid, introduced steady state simulation software in the fourth year course of the chemical engineering degree,[44] proposing a case study on the reactive distillation for the production of MTBE. Simulation in chemical plants educational courses guides students through the principles of unit operations, and is a support in the development of a whole process.[45] For example, simulations reveal theoretical trends of processes, verify the kinetic and thermodynamics of reactions, and calculate fluid phase mixing and equilibria.[42] When proposing experimental activities in parallel with theoretical lectures and simulation, a creative environment that promotes team work is generated.[46]

Process simulation requires a solid theoretical background, as well as a basic knowledge of the computational methods the software are based on. For this reason, lectures on the simulation suite and capabilities are propaedeutic to understand and operate correctly.

Dynamic simulation is required for control strategies, for start up and showdown operations, and for operator training by plant virtualization.[47] The detailed representation of commercial control algorithms, the high-realistic models of several unit operations and the reproduction of the real behaviour of pipes, vessels, and valves are the basis of the most advanced dynamic simulation tools.[43] These options allow students to interact with a virtual plant and to operate instruments. There, the students engage at a practical level (starting up of a pump, sampling, reading gauges, etc.) and in plant management and control. This immersive simulation software combines augmented reality, computational dynamics and computer graphics.[48]

Augmented reality simulation examples concerning simulated chemical plants are rare, while robotics and physical sciences already developed commercial applications and university courses,[49] Nevertheless, new educational proposals are on-going, as for example Eye4Edu project at the University of Milano (Italy). Eye4Edu applies EYESIM software, from AVEVA, that proposes the immersive virtualization of a Crude Distillation Unit (CDU) plant with the representation of the detailed realistic 3D graphics of the whole plant

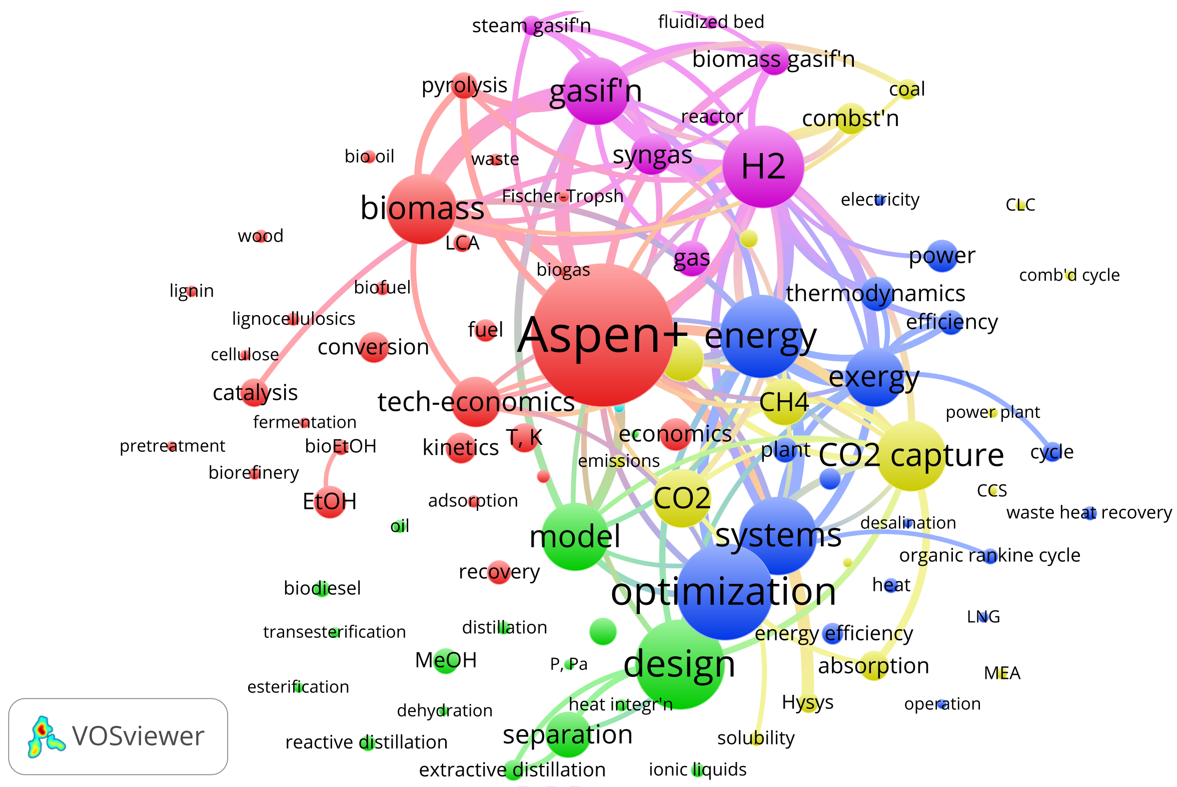

FIGURE 5 VOSViewer keyword bibliometric map based on keywords (Aspen+, Aspen plus, Hysys, and Pro/II) derived from articles that WoSof Science indexed from 2017 to April 2020.[25, 26] The text font size and diameter of the circles are proportional to the number of occurrences in the 2583 articles indexed during this time. Aspen+ (largest circle) appeared in 433 articles. The smallest circles appear in 26 articles.

(Figure 6).[50] DYNSIM software simulates the plant and supplies the physico-chemical properties. The main problems and limitations of the software in education are: 1) the imperfect reproduction of the reality, i.e. there is always some discrepancy compared to the real world. It is therefore impossible to recreate all the layouts of a real plant; 2) simulators are expensive and require maintenance and constant updates; and, 3) operators and teachers need training. Process simulators also train senior engineers and field operators.[51–53] Commercial software simulate mostly reactions, separations, design heat integration and offer a variety of ideal reactors.[54, 55] Depending on the field of application, the software is integrated in the design of a new process, in the simulation of an existing plant, or both.[56] For instance, if we simulate and benchmark the energy consumption of an existing plant, we increase its overall efficiency and save money. In 2017 and 2018, Panjeshahi et al. demonstrated the efficacy of the process integration in the cement[57] and the petrochemical industry,[58] with a total duty saving of 24 % and 14 %, respectively. Cold and hot utilities consumption, energy requirements and emissions of oil & gas,[59, 60] electric power,[61] biofuels,[62] chemical, pharmaceutical and urban systems[63] existing plants can all be reduced with a retrofit based on a simulated process. Optimize a process, instead, requires setting economic, process or environmental objective functions to satisfy. Commercial software, such as AspenPlus, Hysys, PRO II or SuperPro Designer, are designed to work with a fully defined problem (sum of degrees of freedom equal to 0).

FIGURE 6 Operator Training Simulators (OTS) are an educative tool for master and undergraduate students. A dynamic simulation of the plant calculates the consequences of each operator's actions and the student learns through experience.

3.1 | Cost estimation

The log of the investment (log I) of a process unit and even an entire plant is linearly correlated with the log of its capacity (log Q). The slope of the curve depends on the equipment type, specifically on its characteristic dimension,[64] and it varies from 0.33 to 1.0 for equipment and from 0.38 to 0.90 for plants, averaging to a value of 0.6[65] (Eq. 10).

$$I_2 = I_1 \left(\frac{Q_2}{Q_1}\right)^{0.6} \tag{10}$$

For modular units, the learning elasticity is proportional to this exponent and the cost of these units decreases with multiple units—learning/experience.[66] Lang elaborated a simplified method to calculate the total installed cost of equipment based on factors that vary according to the type of plant (solid handling, solid-fluid, or fluids).[67–69]

A more detailed approach, proposed by Guthrie, estimates the bare module cost of a unit, C_{BM}, based on a characteristic variable, A, (volume for a reactor or a column, surface for an heat exchanger, etc.) corrected with pressure coefficients, C_i and material coefficients, B_i (Eq. 11).[70, 71]

$$\begin{cases} C_{BM} = C_p^0(B_1 + B_2 F_M F_P) \\ \log_{10} C_p^0 = K_1 + K_2 \log_{10} A + K_3 \left(\log_{10} A\right)^2 \\ \log_{10} F_P = C_1 + C_2 \log_{10} P + C_3 \left(\log_{10} P\right)^2 \end{cases} \tag{11}$$

F_M the material factor, that is unity for carbon steel and > 1 for other materials, and F_P is a pressure factor. In the case of vessels, a different equation calculates $F_{P,vessel}$ (Eq. 12)

$$\begin{cases} F_{P,vessel} = \dfrac{\dfrac{(P+1)D}{2(850 - 0.6(P+1))} + 0.00315}{0.0063} \\ F_{P,vessel} = 1 \quad \text{if} \quad t_{vassel} < 6.3\,\text{mm} \\ F_{P,vessel} = 1.25 \quad \text{if} \quad P < -0.05\,\text{MPag} \end{cases} \tag{12}$$

which is valid for a corrosion allowance of 3.15 mm.

Ulrich and Vasudevan improved the method of Guthrie and extended his correlation to 58 unit operations of the chemical industry and updated the correlations with a larger database.[72, 73] All the estimations are then actualized by an inflation index like the Marshall & Swift equipment Cost Index, the Chemical Engineering Plant Cost Index (CEPCI), the Engineering News-Record (ENR) construction Index, or the Nelson-Farrar Refinery Construction cost Index (NFRCI), where CEPCI and NFRCI refer to the entire plant and the others to either plant or equipment.[74] Simulation packages have tools to estimate a units' costs, that compute a combination of the methods reported with vendors' quotes/estimates.[75] They prefer to implement algorithm-based methods (Guthrie, Ulrich, Seider[76]) rather then cost-to-capacity or the Lang-based methods. For instance, Aspen Capital Cost Estimator (ACCE)—formerly APEA, the in-house Aspen estimation software, couples design and equipment cost modules, with proprietary time-proven, industry-based direct and indirect factors based on company project history, vendor quotes. These factors are updated regularly.[76] Industry and academia developed dozens of integrated and stand-alone software to estimate cost. Cleopatra and ACCE are the built-in software of PRO/II and AspenOne, respectively. Another example is the built-in Capital Cost Estimator of the CAPE-OPEN project DWSIM. EstPro is a stand-alone process plant cost estimation package from Gulf Production while CapCost, CCEP and DFP are included with books.[74, 76, 77] EconExpert is web-based software. Even software heavily based on updated vendor quotes have errors of at least 25 %.[75] We recommend to operate with one model (or software) to evaluate different plant designs to generate comparable results.

3.2 | Optimization

Most process synthesis problems have multiple feasible solutions. Quantitative optimization techniques select the best of these solutions, which corresponds to a process configuration that maximizes a problem-specific measure of process performance. Any process synthesis problem can be reformulated as an equivalent optimization problem (Eq. 13) composed of three principal components: an objective function, $f(\boldsymbol{x}, \boldsymbol{y})$—a problem-specific measure of process performance, some decision variables, continuous $\boldsymbol{X}$ (process variables), or binary $\boldsymbol{y}$ (they specify the presence or absence of certain process units), and a set of constraints, inequality $g(\boldsymbol{x}, \boldsymbol{y})$ (e.g., minimum product purity, maximum allowable equipment cost, environmental emission limits) and equality $h(\boldsymbol{x}, \boldsymbol{y})$ (e.g., mass, momentum and energy balances, mass and heat transfer correlations, and phase equilibria). The numerical solution of this optimization problem requires discrete or continuous solvers.

$$\begin{cases} \text{Minimize} \quad f(\boldsymbol{x}, \boldsymbol{y}) \\ \text{Subject} \quad \text{to} \quad g(\boldsymbol{x}, \boldsymbol{y}) \le 0, \quad h(\boldsymbol{x}, \boldsymbol{y}) = 0 \\ \boldsymbol{x} \in \boldsymbol{X}, y \in \{0, 1\} \end{cases} \tag{13}$$

Examples of common objective functions include total cost, profit, energy expenditure, exergy loss, environmental and/or social impact, and the error over time of a control model. When two or more conflicting objective functions must be maximized/minimized simultaneously, a dedicated optimization method solve this multi-objective optimization problem. For example, Patel and Padhiyar[78] solved a bio-reactor design problem, in which they simultaneously minimized the batch time and maximized the process yield.

Additionally, Eq. 13 may have multiple solutions, called local optima, and finding the best of these, i.e. the global optimum, is essential. In this case, the user selects special

For Peer Review

types of optimization algorithms that offer global optimality guarantees:

- Linear problems (LP): linear objective function and constraints, and continuous decision variables,
- Mixed-integer linear problem (MILP): linear objective function and constraints, and both discrete and continuous decision variables,
- Non-linear problems (NLP); non-linear objective function and/or constraints, and continuous decision variables, and
- Mixed-integer non-linear problems (MINLP): non-linear objective function and/or constraints, and both discrete and continuous decision variables.

For Peer Review

Scheduling, supply-chain and planning problems are usually LPs or MILPs. NLPs and MINLPs characterize dynamic optimization, design and process synthesis problems. Gradient-based and derivative-free algorithms solve NLPs and MINLPs. Both the algorithms are iterative: they start from an user-supplied initial guess and progressively move towards a solution of the optimization problem (a set of optimal values of the decision variables). Gradient-based methods rely on the values and the derivatives of constraints and objective functions while derivative-free methods rely on the value of the functions. Gradient-based algorithms are suitable for mildly non-linear optimization problems with smooth objective functions and constraints, given that the initial estimates are coherent. Derivative-free methods are appropriate for highly non-linear small-scale optimization problems with both smooth and non-smooth objective function and constraints, even if a good initial guess is unknown. These optimization algorithms are partially complementary, which is the reason why process simulation packages implement a few gradient-based and derivative-free methods, and let the user excogitate on the one to select.

The LP simplex method for LPs,[79] sequential quadratic programming (SQP) for NLPs,[80] and the branch & bound (BB) method for MINLPs are examples of gradient-based methods.[81] SQP is a quasi-Newton algorithm that is computationally efficient but it relies on first and second order derivatives of the objective function and constraints of the optimization problem. For this reason, SQP solves smooth problems best such as LPs and NLPs (Aspen Plus and PRO/II feature SQP as default optimization method). Additionally, SQP does not guarantee convergence to the global optimum.

Derivative-free optimization strategies include BOX, which is a basic implementation of the Nelder-mead simplex algorithm,[82] and COMPLEX, whose implementation details are property of Aspen HYSYS and Aspen Plus. These built-in algorithms handle simple optimization problems with a single objective function, provided that a decent initial guess is available.[83, 84] However, complex or multi-objective optimization problems may require external optimization packages (e.g., CPLEX and XPRESS for solving LPs and MILPs, CONOPT, IPOPT and KNITRO for solving NLPs, and DICOPT, BARON and ANTIGONE for solving MINLPs), interfaced with the simulation flowsheet. For instance, Galli et al.[85] applied Monte Carlo optimization of operating costs and environmental impact of a plant for producing oxygen-enriched air, simulated with PRO/II). Quiroz-Ramírez[86, 87] optimized a bio-butanol plant with AspenPlus, MATLAB, and a home-made VBA script. Eslick et al.[88] optimized an amine absorption process, with Excel, Aspen HYSYS, and the solver NSGAII. Ponce-Ortega et al.[84] developed a procedure to tune any commercial simulator to any multi-optimization algorithm with MS Excel. In fact, they implemented a hybrid stochastic algorithm called I-MODE to maximize the gross annual profit and to minimize the CO_2 emissions of a power and a biodiesel plant.[89] Interfacing commercial simulation packages with external optimization tools is a considerable task with respect to computational power, and time, so we recommend this technique only as a last resort.

3.3 | Energy Integration Intensification

Since 1970,[90] pinch analysis has guided the placement of heat exchanges and other unit operations (distillation, for example) to minimize energy consumption.[9] We visualize each non-reactive stream by a heat content H (kW)—temperature couple. When a differential heat flow dQ is added to a stream, it increases its enthalpy H by $C_P dT$:

$$Q = \int_{T_1}^{T_2} C_P \, dT = C_P(T_2 - T_1) = \Delta H \qquad (14)$$

where T_2 and T_1 are the target and the supply temperature, respectively. In this way, plotting the temperature in terms of the heat content, we have, for each stream, a curve whose slope is $1/C_P$. For a single couple of hot-cold streams this approach is of little use, because to a k hot load increase, corresponds a k cold required duty increase.[91] However, the graphical representation comes in handy when there are several streams involved. In this case:

- **First** plot all the streams separately on a T/H graph. Since we are interested in the ΔH, we only have to place the stream on the right interval of the y axes.
- **Then** add up all the cold and heat duties separately. Since the slope of each stream represents its C_P, whenever two or more streams share a temperature interval $T_1 - T_2$, the heat available in this interval will be $(C_{P,i} +$

$C_{P,ii} + \dots C_{P,n})(T_1 - T_2)$. The outcome of this procedure are the hot and cold composite curves.

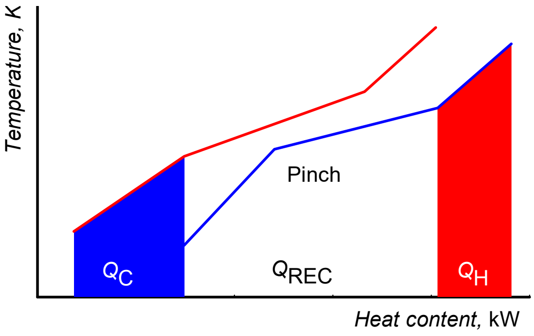

FIGURE 7 Example of hot (red) and cold (blue) composite curves. The blue area is the minimum cooling duty required Q_c and the red area is the minimum heating duty required Q_h.

For Peer Review

When we plot these together, we identify (Fig.7):

- The minimum cooling duty required Q_C.
- The minimum heating duty required Q_H.
- The recovered duty where the curves overlap Q_{REC}.
- The point of closest approach, the pinch point.[90].

This means that as long as the heat exchange across the pinch operates at the corresponding minimum ΔT, it is possible to design an heat exchange network (HEN) who will recover Q_{REC}. The only external heat for the process will only be Q_H and Q_C. Lindhoff et al.[92] expanded the graphical concept of the composite curve, to the *Problem Table* method. This latter method, algebraically splits the process in a cascade of temperature intervals. To do so, we adjust the hot or cold stream temperatures of a negative or positive fraction of ΔT_{pinch}. For instance, we either shift the hot streams by -0.5ΔT_{pinch} and the cold streams by +0.5ΔT_{pinch}, or the hot streams by $-\Delta T_{\text{pinch}}$ and leave the cold strea,s untouched. Setting the temperatures this way guarantees that each interval will either have a positive or a negative duty balance. Foreach interval i we will therefore have a balance:

$$\Delta H_i = (T_i - T_{i+1})(\Sigma C_{\text{P,h}} - \Sigma C_{\text{P,c}})_i \qquad (15)$$

When we sum up the ΔH for all the intervals, we obtain a positive and a negative heat residual. The only way to close the energy balance is to introduce a cold and a hot utility. We have to supply a Q_C and Q_H at a thermal level lower and higher of the lowest and highest interval of the cascade. That is, if the highest interval is at $T = 160\,°\text{C}$, we include a hot utility above 160 °C. Brown et al.,[93] demonstrated how pinch technology achieved an average of 25 % cut on the site energy bills, equivalent to 30 million pounds in 1989, for all the main sectors in chemical engineering. Although other methods like the composite curve method are equivalent,[76, 94] the inherent modularity of the Problem Table method makes it perfect to implement it in process simulators.

Process simulators perform pinch point analysis (PPA) to design an optimal heat exchange network (HEN), minimize the energy requirement of the process and the utilities consumption.[54] Several software packages on the open market incorporate a PPA section such as HYSIM, HEXTRAN, ADVENT by Aspen Tech, TARGET or, more recently, FI^{2}EPI.[95]. Simulators couple PPA analysis with a UA heat exchange model, in this way they work on the design of the network outside the simulation environment. They share the possibility of importing the H&M balance results from outside, and the presence of at least a manual and an automatic design. However, we recommend completing individual PPA before integrating them all into the entire network. In fact, the automatic mode will achieve the target (minimum number of heat exchangers or maximum energy saving), regardless the process engineering of the system. For instance, it might suggest you to exchange between stream 1 and stream 1000 of the plant, not considering the physical distance. Or it might suggest to split one stream into five sub-streams, to maximize its heat exchange. Or, based on what you are looking for, it might propose uneconomical heat exchangers. On the other hand, this software will not suggest you where to locate an equipment with respect to the pinch.[9, 76] Furthermore, energy intensification works as long as the plant operates at steady-state. However, while the steady-state defines the baseline operation,[96] in real life a plant alternates between steady-state and unsteady state (start-up, shut-down, feed composition variations, for example).[97–99] For this reason, we recommend understanding the dynamics of all the operations in the simulation before integrating energy.

3.4 | Process simulation as research tool

Process simulators started as academical tools but soon thereafter industry adopted this software for design. Now this tool is returning to academia and research: Saidi and Kadkhodayan[100] integrated an experimental Taguchi optimization method to a process simulation to bridge industrial operation with laboratory scale experiments. Jafari simulated a circulating fluidized bed with a sequential modular approach and showed that it could be added to a commercial simulation.[101] Commercial software developers are expanding their offering with non conventional components, such as electrolytes, solids, or plastic[56]. Process simulators support the experiments and approximate thermodynamic parameters with equations of state for those parameters difficult to measure. Shen et al.[102] studied a primary cooler for a Coke Oven Gas LiBr heat pump. They relied on simulation results to estimate operating parameters they were unable to measure experimentally to build a refrigerator pilot plant. They retroactively validated the simulation with pilot plant data and reported an error of 0 % to 10 % depending on the parameter. Process simulators are also tools to calculate thermodynamic properties—enthalpy, entropy, Gibb's free energies, densities, viscosity—as input to estimate reaction kinetics, heat duties, and phase changes. analysis of a reaction. For example, to define an experimental plan to study methane partial oxidation, we estimate the theoretical bound on conversion and selectivity as a function of temperature and pressure, then determine the relationship between coking and oxygen partial pressures. These data define the boundaries of the plan. PRO/II simulated a Gibbs reactor (RGibbs) with the Peng-Robinson equation of state. A case study varied the O_2/CH_4 ratio in the range 0.2 to 1.0 and the temperature of the reactor from 1000 K to 1200 K (Figure 8).

For Peer Review

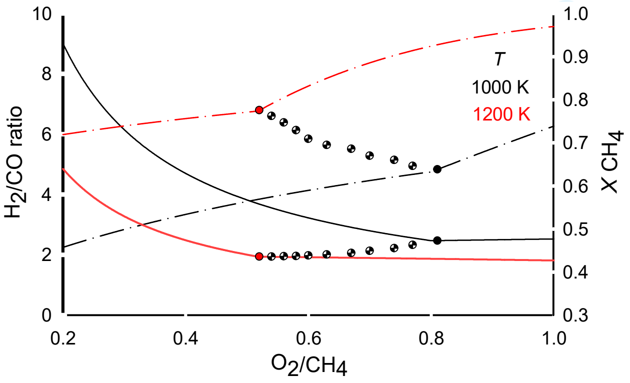

FIGURE 8 Thermodynamic (maximum) methane conversion (dotted lines) and obtainable H_2/CO ratio (full lines) for the catalytic partial oxidation of methane at 1000 K (black lines) and 1200 K (red lines). The simulator also identifies at which ratio coke starts to form (points). We calculated the H_2/CO ratio at which coke forms every 20 K. The simulations took less than 8 min to complete.

4 | UNCERTAINTY

4.1 | Limitations

Commercial software licenses are expensive and to simulate non-standard operation often additional code is required. The commercial software models and algorithms are robust, and most of the errors come from user-added inputs. Integrating custom code into process simulators is cumbersome and takes longer than writing a program from scratch.

To set up a simulation file, the following procedure applies:

- Define the units of measure, according to a predefined set (metric, english or SI) or to a customized set,
- Select the components (from a databank),
- Choose a thermodynamic model,
- Check the simulation tolerances, and
- Build the flowsheet. In case of recycles see Section 2

Errors compound as the simulations proceeds from one step of the process to the next. Simulators have no error assessment capability.

4.2 | Sources of error

4.2.1 | Units of measure (UOM)

One of the most frequent error source is incorrect data entry, i.e. the human error. Even though most of the commercial simulation suites have a window dedicated to the selection of UOMs, it is possible to change these inside all unit operations. For example, if the ENGLISH system is set as default, but for a specific distillation application the user knows that the distillate flowrate is 100 kmol h^{-1}, inserting a value of 100 without correcting the UOM will result in setting a distillate flow of 100 klbmol h^{-1}, resulting in a error of 1/2.2. The most likely error is the simulation will not converge to a solution.

Another common mistake is when users define units of measure in the reaction kinetics window. The user inputs the main kinetic parameters—Arrhenius constants, A_{pre}, activation energies, E_a,and reaction orders, ϵ—for a kinetic model or by defining a pseudo-homogeneous kinetic scheme (Eq. 16):

$$r = A_{\text{pre}} \exp\left[-\frac{E_a}{R}\left(\frac{1}{T} - \frac{1}{T_o}\right)\right] \cdot T^n \cdot \prod a_i^{\epsilon} \qquad (16)$$

where a the activity of reagents and products (calculated as the product concentration, molar fraction or according to a thermodynamic model).

PRO/II permits the user to supply the values of these parameters and the volume and the pressure UOMs, as the r is always defined as (mol Volume^{-1} Time^{-1}). Also, the value of the gas constant R is 8.314 J mol^{-1} K^{-1}. Therefore, the UOMs of pressure and volume in the reaction window must be chosen to give J when multiplied, i.e. or kPa and L, or Pa and m^3, respectively.

On the other hand, when defining a kinetic procedure, the user writes a FORTRAN or pseudo-FORTRAN code and calculates the system of differential equation that the software will integrate using a numerical method (typically Runge-Kutta). In this case, the user should provide the proper rate expression UOM to the solver, in mol Volume^{-1} Time^{-1}.

4.2.2 | Components selection

The selection of components means, for the software, to import all the pure component properties, the fixed (molecular weight, acentric factor, normal boiling point, critical point, Van der Waals area and volume, etc.) and the temperature dependent properties (density, vapor pressure, viscosity, heat capacity, etc.).

We always recommend to employ components present in the software database, as these are usually updated with one or more methods to estimate the component's properties. Besides, a good practice is to operate with less than 40 pure compounds per simulation.

In the case of new components (those not yet in any database), all the physical properties should be provided as well; we recommend to retrieve the physical and thermodynamic properties on databases such as NIST chemistry WebBook [103] or Knovel. [104]

There are dozens of equation to estimate thermodynamic properties. [105] This reference covers also electrolytes systems and solids solubilities. We recommend to pay particular attention while operating with new materials and processes such as:

- Green and biological processes,
- Molten organic and inorganic salts
- Nonconventional solids (i.g. solid with unknown chemical formula)
- Electrolytes
- Novel materials (nanomaterials, composite materials, copolymers, etc.)

Commercial software as ASPEN, PRO/II and ProSim estimate new component's properties from UNIFAC group contribution methods. [106] In this case, the flaws of the simulators rely in the estimation of the physical properties, which are either imprecise, [107] or incomplete.

4.2.3 | Thermodynamics

One of the most critical points is the selection of a proper thermodynamic method to calculate mixture properties (excess Gibbs free energy or fugacity). Analyzing the system (Fig. 9) leads to the selection of a $\gamma - \phi$ or a $\phi - \phi$ approach. However, this is a simplified scheme as each component mixture may show different non-idealities. For example, the system water and acetic acid (fully miscible, condensed phase and no supercritical gases dissolved) should be treated with an activity coefficient model. However, with UNIQUAC equations and the binary parameters available in PRO/II database only, the software calculates an non-existent azeotrope (Fig. 10). Only adding a correction for the gas phase association of acetic acid (Hyden-O'Connel equation [110]) or regressing with a robust method experimental data led to a sound simulation. [28] Therefore, we always recommend a literature analysis of the phase equilibria of the systems involved. Gmehling et al. [111] published a book regarding chemical thermodynamics and process simulation. Gani and O'Connel [108] created a decision matrix to select proper physical methods and thermodynamic models, depending on the unit operation simulated and the operative parameters.

4.2.4 | Tolerances

Similar to any other iterative calculation algorithm, the solver of a process simulator needs specific tolerances to terminate the calculations. Generally the default parameters (1×10^{-3} for temperature and pressure —relative), 1×10^{-6} for compositions (absolute), and 1×10^{-4} for duties—relative) are sufficient to guarantee the robustness of the termination criteria.

4.2.5 | Flowsheet

Any unit operation requires specifications to close the mass and energy balances. Generally these specifications are temperatures, pressures, or material flows (like the flow of distillate and residue for a distillation column) but any unit may have peculiar specifications (reactor dimensions for a plug flow or time for a batch reactor, reflux ratio for a distillation column, etc.). Converging to a viable solution depends on setting these specifications correctly. Most of the error in this case are related to the unit of measures of the specifications or to gross errors such as setting a too high reflux ratio or a distillate flow higher than the feed flow rate of the column.

Some specifications make the solver's life harder. If the simulation never converges to a solution, the user should replace some of them with equivalent specifications (e.g, instead of decreasing the molar fraction of impurities in the distillate, increase the reflux ratio until the composition reaches the desired value). Another common mistake is to set the composition of distillate or residue to a value impossible to obtain with the number of theoretical trays specified (such as imposing a distillate or residue purity higher than the minima/maxima azeotropic compositions) for distillations, or imposing by mistake to concentrate the heavy component in the distillate rather than in the residue stream.

If the user imposes extreme constraints (i.e. a separation in which the molar fraction of impurities in the product is less than 1 ppm), then they should start with a more relaxed constraint (±5 %) and tighten it up gradually.

For Peer Review

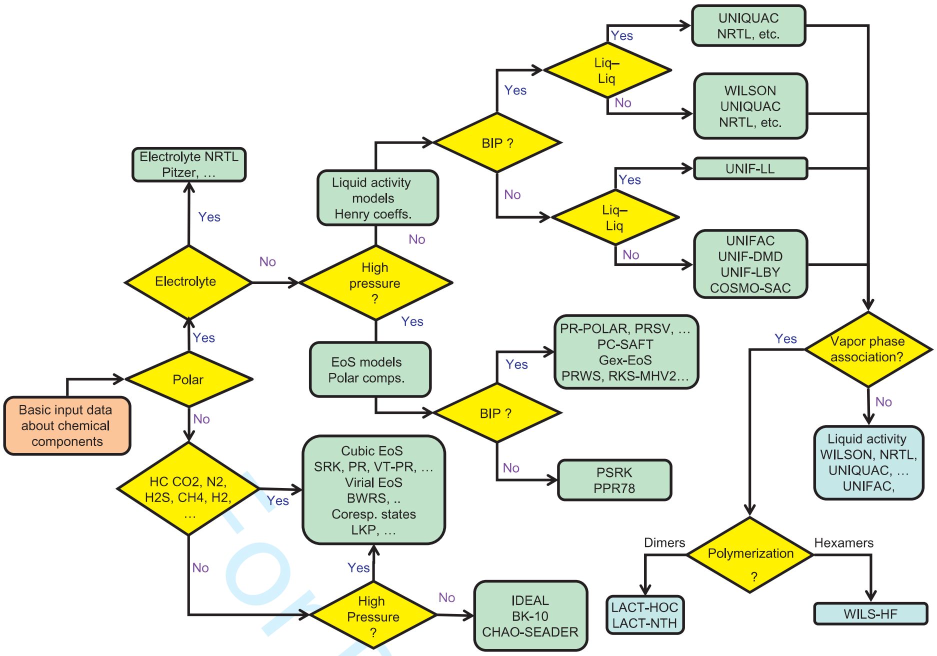

FIGURE 9 Simplified decisional flow diagram for the selection of a thermodynamic method (Tab 2).[108, 109]. *Reprinted from Computer Aided Chemical Engineering, Volume 13, Second Edition—Integrated design and simulation of chemical processes, Alexandre C. Dimian, Costin S. Bildea, Anton A. Kiss, Chapter 6: Phase equilibria, 248, Copyright (2020), with permission from Elsevier.*

For Peer Review

TABLE 2 Most employed thermodynamic models to calculate components' fugacities or activity coefficients.

Model	Advantages	Limitations
	Equation of State (EoS)	
IDEAL	Good for mixtures of similar fluids at low P and T	$P < 0.3$ MPa
Chao-Seader[112]	Generalized correlation for hydrocarbon mixture	$P < 14$ MPa, $T < 260\,°C$ Dissolved gas < 20 % by mole
Lee-Kesler[113]	Correlation for $\bar{H}$, $\bar{S}$ and $\bar{\rho}$	Not good for highly polar mixtures
Redlich-Kwong[114]	Calculates hydrocarbons VLE accurately	Not good for polar compounds
Peng-Robinson[115]	Better than SRK with polar components	Worse for hydrocarbonsmixtures
UNIWAALS[116]	Predictive method	Components' liquid volumes required
	Activity coefficients	
Wilson[117]	Polar or associating components	Does not calculate LLE
NRTL[118]	Strongly nonideal mixtures	Very different molecules dimensions
UNIQUAC[119]	Very good with most of systems	No electrolytes
UNIFAC[120]	Predictive method	$T << T_C$, $P < 8.5$ MPa

4.2.6 Initialization

All numerical methods for the resolution of unit operations require initial values to start iterating. However, all commercial packages possess several initialization methods for each

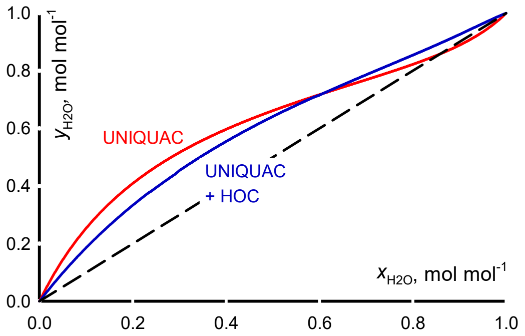

FIGURE 10 The correct selection of a thermodynamic method (blue line) avoids the calculation of non-existent azeotropes (red line) for the binary water-acetic acid system, $P = 101.325\,\text{kPa}$. UNIQUAC parameters: $a_{\text{ij}} = -118.309$, $b_{\text{ij}} = -0.124\,345\,\text{K}^{-1}$, $a_{\text{ji}} = 402.126$ and $b_{\text{ji}} = -0.355\,725\,\text{K}^{-1}$.

unit operation, and these algorithms vary from one unit to another. For an exhaustive description of each method consult the software manual.

For complex simulations, involving several units and recycling streams, we recommend choosing initial conditions judiciously, or to estimate them based on shortcut calculations or literature data, to facilitate the algorithm convergence to the solution. Poor initial guesses will increase the number of iterations unnecessarily (if the simulation converges at all)..

To initiate the initial guess for tear streams, a good strategy is to simulate the flowsheet with no recycle, and then connect the recycles, so that the software stores after the first simulation the initial values of flow rate, temperature, pressure, and compositions of these streams.

4.3 | Sources of error in the EO Approach

While thermodynamic, component selection, and flowsheets are common for every computational architecture, the EO is intrinsically free of any sources of error related to iteration. However, since the EO solves all the units at the same time, the initial values are essential. As a rule of thumb, we recommend to first run the simulation in a SM environment, and then implement the SM outcome as initial value for the EO. This doesn't mean that the SM needs to converge, but at least each block has to be solved once. The closer the SM solution is to the real solution, the easier it will be for the EO to converge without error. While this initialization technique is the one currently present in commercial simulators, new approaches such as the *graphic approach*,[121] or the *pseudo-transient*,[122] are recently emerging.

5 | CONCLUSIONS

Engineers design, control, optimize, retrofit, and estimate the cost of a plants and equipment with process simulators, which calculate thermodynamic properties, model vapor- and/or liquid-liquid equilibria of mixtures and close mass and energy balances. Process simulation will become ever more present in a chemical engineer's arsenal of tools to address society's desire for sustainable products that minimize waste and maximize recyclability as the mathematical methods and computational power improve such that any PC can run these multi-objective optimization problems. The feedstocks of the future will become more heterogeneous (bio-based, waste-based, for example) and because of this complexity, new models, unit operations and thermodynamic correlations will be needed. Quantification of the uncertainty on model predictions is another key aspect that should be further improved in future simulators as well as adding stochastic optimization capabilities to these tools.

Prospectives for future innovations include:

- generating kinetics databases (including packages like Chemkin;[123]
- handling solids and other unconventional materials (plastic mixtures, urban waste, sludge, and poorly characterized complex mixtures);
- better dynamic simulators that consider the time scale of reaction versus hydrodynamics contribution;
- new operating functions for process intensification[124, 125] of current processes;
- more reliable safety assessments that autiomatically identify hazardous operating conditions (HAZOP automatic calculation of lower and upper explosion limits,[126] identification of risk zones, improving the hazard identification when streams mix);
- integrating computational fluid dynamics;
- dynamically query other databases on the internet (NIST);
- integrating artificial intelligence and artificial neural network algorithms for control and optimization;
- improving the thermodynamic packages for new molecules (pharma and green chemistry);
- improving the predictive control of unit operations;
- expanding virtual reality that reproduces realistic plant dimensions; and,

For Peer Review

- improving the integration between home-made code and process simulators.

Also, the integration of multiobject optimization on performance key indicators not only based on economic parameters, such as the CO_2 equivalent or the water footprint of a process (environmental impact) or the hazard and operability analysis, which at the moment is feasable via open platform communication.[127].

As stated by John E. Coon et al. in 1998 "*It is the **responsibility** of the engineer to validate the quality of process design, troubleshoot, and optimization results from simulation tools*" and that wrong answers are "*the results of getting the correct answer to a poorly chosen question*".[128]

For Peer Review

NOMENCLATURE

A	Characteristic variable of an equipment
$\boldsymbol{a}$	Optimal step length
ACCE	Aspen capital cost estimator
A_{pre}	Arrhenius pre-exponential factor
a_i	Activity of component i
$B()$	Jacobian matrix estimation
B_i	Material coefficients
C_{BM}	Cost of bare module
CCS	Carbon capture and sequestration
CDU	Crude distillation unit
CEPCI	Chemical Engineering Plant Cost Index
C_i	Pressure coefficients
C_{P}	Specific heat
D	Diameter of a vessel
$\boldsymbol{d_n}$	search direction vector
E_{a}	Reaction activation energy
ENR	Engineering news-record
EO	Equation oriented approach
EoS	Equation of State
F_{M}	Material factor
F_{P}	Pressure factor
$\bar{H}_i$	Enthalpy of component i in mixture
HOC	Hyden-O'Connel
I	Investment
$J()$	Jacobian matrix
K_i	Cost coefficient of an equipment
LLE	Liquid-liquid equilibrium
LP	Linear problem
MILP	Mixed integer liner problem
MINLP	Mixed integer non-liner problem
MTBE	Methyl-tertbutil ether
NFRCI	Nelson-Farrar Refinery Construction cost Index
NLP	Non-linear problem
OTS	Operator training simulator
P	Pressure
PPA	Pinch point analysis
PR	Peng-Robinson
Q	Plant capacity
Q_{c}	Minimum cooling duty
Q_{h}	Minimum heating duty
Q_{rec}	Recovered duty
R	Gas constant
r	reaction rate
$\bar{S}_i$	Entropy of component i in mixture
SM	Sequential modular approach
SQP	Successive quadratic programming
SRK	Soave-Redlich-Kwong
T	Temperature
T_{C}	Critical temperature
t_{vessel}	Vessel wall thickness
TAC	Total annual costs
UOMs	Unit of measures
VLE	Vapor-liquid equilibrium
$\boldsymbol{x_n}$	solution vector at the iteration n
$\boldsymbol{x_s}$	solution vector

Greek Letters

γ	activity coefficient
ϵ	reaction order
ϕ	fugacity
$\bar{\rho}_i$	Density of component i in mixture

ACKNOWLEDGEMENTS

The authors aknowledge professor Rahmat Sotudeh Gharebagh for his precious comments and suggestions, and the fruitful discussions.

References

[1] J. J. McKetta, J. J. McKetta, ed., Encyclopedia of Chemical Processing and Design, CRC Press, New York **1980**, pp. 150 – 458.

[2] D. C. Y. Foo, N. Chemmangattuvalappil, D. K. Ng, R. Elyas, C.-L. Chen, R. D. Elms, H.-Y. Lee, I.-L. Chien, S. Chong, C. H. Chong, Chemical Engineering Process Simulation, Elsevier, Amsterdam, Netherlands **2017**.

[3] G. S. Patience, *Can. J. Chem. Eng*. **2018**, *96*, 2312.

[4] A. W. Westerberg, H. P. Hutchinson, R. L. Motard, P. Winter, Process Flowsheeting, Cambridge Universities Press, Cambridge, UK **1979**.

[5] R. S. Mah, R. S. Mah, ed., Chemical Process Structures and Information Flows, Butterworth-Heinemann **1990**, pp. 125 – 183.

[6] G. Buzzi-Ferraris, F. Manenti, Nonlinear Systems and Optimization for the Chemical Engineer, Wiley-VCH, Weinheim, Germany **2013**.

[7] Aspen Plus 2004.1 Getting Started Using Equation Oriented Modeling **2004**.
URL https://sites.ualberta.ca/CMENG/che312/F06ChE416/HysysDocs/AspenPlus20041GettingStartedEOModeling

[8] H.-S. Chen, M. Stadtherr, *AICHE journal* **1985**, *31*, 1843.

[9] A. C. Dimian, C. S. Bildea, A. A. Kiss, Applications in Design and Simulation of Sustainable Chemical Processes, Elsevier **2019**.

[10] R. Raoni, A. R. Secchi, E. C. Biscaia Jr, *Comput. & Chem. Eng.* **2017**, *96*, 169.

[11] F. Zhao, X. Chen, L. Zhu, *AIChE J.* **2017**, *63*, 2764.

[12] H. A. J. Watson, Robust simulation and optimization methods for natural gas liquefaction processes, Ph.D. thesis, Massachusetts Institute of Technology **2018**.

[13] D. Himmelblau, *Chem. Eng. Sci.* **1967**, *22*, 883 .

[14] H. Weinblatt, *J. ACM* **1972**, *19*, 43 .

[15] R. Barkley, R. Motard, *The Chem. Eng. J.* **1972**, *3*, 265 , an International Journal of Research and Development.

[16] J. H. Christensen, D. F. Rudd, *AIChE J.* **1969**, *15*, 94.

[17] R. S. Upadhye, E. A. Grens II, *AIChE J.* **1975**, *21*, 136.

[18] R. L. Motard, A. W. Westerberg, *AIChE J.* **1981**, *27*, 725.

[19] J. F. Boston, S. L. Sullivan Jr, *The Can. J. Chem. Eng.* **1974**, *52*, 52 .

[20] R. B. Saeger, P. R. Bishnoi, *The Can. J. Chem. Eng.* **1986**, *64*, 759.

[21] J. Jelínek, *Comput. & Chem. Eng.* **1988**, *12*, 195 .

[22] J. H. Wegstein, *Commun. ACM* **1958**, *1*, 9 .

[23] O. Orbach, C. M. Crowe, *The Can. J. Chem. Eng.* **1971**, *49*, 509 .

[24] C. G. Broyden, *Math. computation* **1965**, *19*, 577 .

[25] N. J. van Eck, L. Waltman, *Sci.* **2010**, *84*, 523.

[26] Clarivate Analytics, Web of Science Core Collection **2020**, accessed on 23 April 2020.
URL http://apps.webofknowledge.com

[27] W.-J. Hung, I.-K. Lai, Y.-W. Chen, S.-B. Hung, H.-P. Huang, M.-J. Lee, C.-C. Yu, *Ind. & Eng. Chem. Res.* **2006**, *45*, 1722.

[28] C. Pirola, F. Galli, F. Manenti, M. Corbetta, C. L. Bianchi, *Ind. & Eng. Chem. Res.* **2014**, *53*, 18063.

[29] Y. Xiao, W. Cai, H. Sun, F. Shi, G. Li, *The Can. J. Chem. Eng.* **2018**, *96*, 722.

[30] S. Wu, K. Guo, C. Liu, W. Qi, T. Zhang, H. Liu, *The Can. J. Chem. Eng.* **2018**, *96*, 2431.

[31] C.-K. Kuei, M.-D. Lee, *The Can. J. Chem. Eng.* **1991**, *69*, 347.

[32] N. A. Bahari, W. N. R. Wan Isahak, M. S. Masdar, Z. Yaakob, *Int. J. Energy Res.* **2019**, *43*, 5128.

[33] D. Previtali, M. Longhi, F. Galli, A. D. Michele], F. Manenti, M. Signoretto, F. Menegazzo, C. Pirola, *Fuel* **2020**, *274*, 117804.

[34] B. Duhoux, R. T. Symonds, R. Hughes, P. Mehrani, E. J. Anthony, A. Macchi, *The Can. J. Chem. Eng.* **2020**, *98*, 75.

[35] D. W. Keith, G. Holmes, D. S. Angelo], K. Heidel, *Joule* **2018**, *2*, 1573.

[36] J. C. Goncalves, A. E. Rodrigues, *The Can. J. Chem. Eng.* **2015**, *93*, 2205.

[37] P. Lutze, R. Gani, J. M. Woodley, *Chem. Eng. Process. Process. Intensif.* **2010**, *49*, 547 .

[38] E. Barbera, F. Mantoan, A. Bertucco, F. Bezzo, *The Can. J. Chem. Eng.* **2020**, *n/a*.

[39] K. D. Dahm, R. P. Hesketh, M. J. Savelski, *Chem. Eng. Educ.* **2002**, *36*, 192.

[40] A. M. Borreguero, J. L. Valverde, J. M. García-Vargas, L. Sánchez-Silva, *Comput. Appl. Eng. Educ.* **2019**, *27*, 1267.

[41] S. Yang, L. Yang, C. He, *Process. Saf. Environ. Prot.* **2001**, *79*, 329 .

[42] C. Pirola, *Educ. for Chem. Eng.* **2019**, *28*, 54 .

[43] T. M. Komulainen, R. Enemark-Rasmussen, G. Sin, J. P. Fletcher, D. Cameron, *Educ. for Chem. Eng.* **2012**, *7*, e153 .

For Peer Review

[44] L. Calvo, C. Prieto, *Educ. for Chem. Eng.* **2016**, *17*, 65 .

[45] A. Rafael, F. Bernardo, L. Ferreira, M. Rasteiro, J. Teixeira, *Educ. for Chem. Eng.* **2007**, *2*, 20 .

[46] L. Fregolente, H. Venturelli, J. Rodrigues, E. Da Silva, I. Diniz, R. Wolf Maciel Maria, *Chem. Eng. Transactions* **2018**, *69*, 427.

[47] S. Yang, L. Yang, C. He, *Process. Saf. Environ. Prot.* **2001**, *79*, 329 .

[48] R. M. Ghoniem, H. Abas, H. Bdair, *Procedia Comput. Sci.* **2018**, *135*, 490 .

[49] C. Foreman, M. Hilditch, N. Rockliff, H. Clarke, A Comparison of Student Perceptions of Physical and Virtual Engineering Laboratory Classes, Springer International Publishing, Cham **2020**, pp. 151–167.

[50] C. Pirola, F. Galli, C. Peretti, *Comput. & Chem. Eng. J.* **2020**, accpted.

[51] P. R. C. Camargo, E. de Barros Basso, FCC Training simulator–Petrobras experience **2005**.

[52] J. Isimite, F. Baganz, V. C. Hass, *J. Chem. Technol. & Biotechnol.* **2018**, *93*, 1529.

[53] D. S. Patle, D. Manca, S. Nazir, S. Sharma, *Virtual Real.* **2019**, *23*, 293.

[54] J. Haydary, Reactors, chapter 5, John Wiley & Sons, Ltd **2019**, pp. 101–124.

[55] S. S. Elnashaie, Modelling, simulation and optimization of industrial fixed bed catalytic reactors, volume 7, CRC Press **1994**.

[56] J. Haydary, Chemical Process Design and Simulation: Aspen Plus and Aspen Hysys Applications, John Wiley & Sons **2019**.

[57] M. A. Mirzakhani, N. Tahouni, M. H. Panjeshahi, *Energy* **2017**, *130*, 382.

[58] M. Sardarmehni, N. Tahouni, M. H. Panjeshahi, *Energy* **2017**, *127*, 623.

[59] M. V. Kanischev, L. M. Ulyev, R. E. Chibisov, M. A. Vasilyev, *Chem. Eng. Transactions* **2018**, *70*, 1099.

[60] Q. Zhang, M. Yang, G. Liu, X. Feng, *J. Clean. Prod.* **2016**, *112*, 4799.

[61] S. S. Chauhan, S. Khanam, *Energy* **2019**, *173*, 364.

[62] R. R. Tan, D. C. Y. Foo, K. B. Aviso, D. K. S. Ng, *Appl. Energy* **2009**, *86*, 605.

[63] G. Venkatesh, *Resour. Environ. Inf. Eng.* **2019**, *1*, 1.

[64] D. I. Garnett, G. S. Patience, *Chem. Eng. Prog.* **1993**, *89*, 76.

[65] R. Williams Jr., *Chem. Eng.* **1947**, *54*, 124 .

[66] G. S. Patience, D. C. Boffito, *J. Adv. Manuf. Process.* **2020**, *2*, e10039.

[67] H. J. Lang, *Chem. Eng* **1947**, *54*, 27.

[68] H. J. Lang, *Chem. Eng.* **1947**, *54*, 130 .

[69] H. J. Lang, *Chem. Eng.* **1948**, *55*, 112 .

[70] K. M. Guthrie, *Chem. Eng.* **1969**, *76*, 114.

[71] K. Guthrie, Process Plant Estimating, Evaluation, and Control, Craftsman Book Company of America **1974**.

[72] G. D. Ulrich, P. T. Vasudevan, Chemical Engineering Process Design and Economics: A Practical Guide, Process Publishing, Duhram, New Hampshire **2007**.

[73] G. D. Ulrich, P. T. Vasudevan, *Chem. Eng.* **2006**, 66 . URL http://terpconnect.umd.edu/~NSW/chbe446/HowToEstimateUtilityCosts-UlrichVasudevan2006.pdf

[74] R. A. Turton, J. A. Shaeiwitz, D. Bhattacharyya, W. B. Whiting, Analysis, synthesis and design of chemical processes, 5th edition, Pearson Education **2018**.

[75] Y. Feng, G. P. Rangaiah, *Chem. Eng.* **2011**, 22 . URL http://sgpwe.izt.uam.mx/files/users/uami/rmro/2122096/S11/Evaluating_capital_cost_estimation_program.pdf

[76] W. D. Seider, J. D. Seader, D. R. Lewin, Product & Process Design Principles: Synthesis, Analysis and Evaluation, (With CD), John Wiley & Sons **2009**.

[77] R. Sinnott, G. Towler, Chemical engineering design: SI Edition, Butterworth-Heinemann **2019**.

[78] N. Patel, N. Padhiyar, *Chem. Eng. Res. Des.* **2017**, *119*, 160 .

[79] Practical optimization: a gentle introduction **2006**. URL https://eclass.gunet.gr/modules/document/file.php/LABGU361/PRACTICAL%20OPTIMIZATION%20GENTLE%20INTRODUCTION%202015.pdf

[80] L. T. Biegler, Nonlinear programming: concepts, algorithms, and applications to chemical processes, volume 10, Siam **2010**.

For Peer Review

[81] R. A. Stubbs, S. Mehrotra, *Math. Program.* **1999**, *86*, 515 .

[82] J. A. Nelder, R. Mead, *The Comput. J.* **1965**, *7*, 308.

[83] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, et al., Evolutionary algorithms for solving multi-objective problems, volume 5, Springer **2007**.

[84] J. M. Ponce-Ortega, L. G. Hernandez-Perez, Optimization of Process Flowsheets through Metaheuristic Techniques, Springer **2019**.

[85] F. Galli, C. Pirola, D. Previtali, F. Manenti, C. L. Bianchi, *J. Clean. Prod.* **2018**, *171*, 147.

[86] J. J. Quiroz-Ramirez, E. Sanchez-Ramirez, S. Hernandez, J. H. Ramirez-Prado, J. G. Segovia-Hernandez, *Ind. & Eng. Chem. Res.* **2017**, *56*, 1823.

[87] J. J. Quiroz-Ramirez, E. Sanchez-Ramirez, S. Hernandez-Castro, J. G. Segovia-Hernandez, J. M. Ponce-Ortega, *ACS Sustain. Chem. & Eng.* **2017**, *5*, 4018.

[88] J. C. Eslick, D. C. Miller, *Comput. & Chem. Eng.* **2011**, *35*, 1488 .

[89] S. Sharma, G. P. Rangaiah, *Comput. & chemical engineering* **2013**, *56*, 155.

[90] B. Linnhoff, J. R. Flower, *AIChE J.* **1978**, *24*, 633.

[91] I. C. Kemp, Pinch analysis and process integration: a user guide on process integration for the efficient use of energy, Elsevier **2011**.

[92] B. Linhoff, J. Flower, *AIChE J.* **1978**, *24*, 642.

[93] K. Brown, Process Integration Initiative: A Review of the Process Integration Activities Funded Under the Energy Efficiency R and D Programme, Energy Technology Support Unit (ETSU) **1989**.

[94] T. Umeda, J. Itoh, K. Shiroko, *Chem. Eng. Prog.* **1978**, *74*, 70.

[95] P. M. Pereira, M. C. Fernandes, H. A. Matos, C. P. Nunes, *Appl. Therm. Eng.* **2017**, *114*, 523.

[96] E. Rosen, ACS Publications **1980**.

[97] N. M. Kazerooni, R. Eslamloueyan, M. Biglarkhani, *Int. J. Ind. Chem.* **2019**, *10*, 107.

[98] Y. Xu, H. Dinh, Q. Xu, F. T. Eljack, M. M. El-Halwagi, *J. Clean. Prod.* **2020**, 120129.

[99] P. Thomas, S. Bhattacharyya, A. Patra, G. Rao, *Comput. & chemical engineering* **1998**, *22*, 1515.

[100] M. Saidi, H. Kadkhodayan, *J. Environ. Chem. Eng.* **2020**, *8*, 103772.

[101] H. Jafari, A. Sheikhi, R. Sotudeh-Gharebagh, *The Can. J. Chem. Eng.* **2020**, *98*, 1003 .

[102] Y. Shen, Z. Tang, D. Guo, C. Wu, D. Wang, A. Jiang, Y. Wen, *Appl. Therm. Eng.* **2019**, *160*, 113983.

[103] NIST Standard Reference Database Number 69 **2020**. URL https://webbook.nist.gov/chemistry/

[104] Engineering Data and Technical References to Solve Engineering Problems **2020**.
URL https://app.knovel.com/web/index.v?jsp=main

[105] B. E. Poling, J. M. Prausnitz, J. P. O'Connell, The properties of gases and liquids, 5th edition, Mcgraw-hill New York **2001**.

[106] T. Jensen, A. Fredenslund, P. Rasmussen, *Ind. & Eng. Chem. Foundamentals* **1981**, *20*, 239.

[107] W. Wakeham, M. Assael, *BULGARIAN CHEMICAL COMMUNICATIONS* **2019**, *51*, 9 .

[108] R. Gani, J. P. O'Connell, *Comput. & Chem. Eng.* **1989**, *13*, 397 .

[109] E. C. Carlson, *Chem. Eng. Prog.* **1996**, 35 .

[110] J. G. Hayden, J. P. O'Connell, *Ind. & Eng. Chem. Process. Des. Dev.* **1975**, *14*, 209.

[111] J. Gmehling, M. Kleiber, B. Kolbe, J. Rarey, Chemical thermodynamics for process simulation, John Wiley & Sons **2019**.

[112] K. C. Chao, J. D. Seader, *AIChE J.* **1961**, *7*, 598.

[113] B. I. Lee, M. G. Kesler, *AIChE J.* **1975**, *21*, 510.

[114] O. Redlich, J. N. S. Kwong, *Chem. Rev.* **1949**, *44*, 233.

[115] D.-Y. Peng, D. B. Robinson, *Ind. & Eng. Chem. Fundamentals* **1976**, *15*, 59.

[116] P. A. Gupte, P. Rasmussen, A. Fredenslund, *Ind. & Eng. Chem. Fundamentals* **1986**, *25*, 636.

[117] G. M. Wilson, *J. Am. Chem. Soc.* **1964**, *86*, 127.

[118] H. Renon, J. M. Prausnitz, *AIChE J.* **1968**, *14*, 135.

[119] D. S. Abrams, J. M. Prausnitz, *AIChE J.* **1975**, *21*, 116.

For Peer Review

[120] A. Fredenslund, R. L. Jones, J. M. Prausnitz, *AIChE J.* **1975**, *21*, 1086.

[121] S. T. Holland, R. Abbas, D. Hildebrandt, D. Glasser, *Ind. & engineering chemistry research* **2010**, *49*, 327.

[122] R. C. Pattison, M. Baldea, *AIChE J.* **2014**, *60*, 4104.

[123] Chemistry Effects Predicting Simulation Software **2020**.
URL https://www.ansys.com/products/fluids/ansys-chemkin-pro

[124] D. F. Rivas, D. C. Boffito, J. A. F. Albanese, J. Glassey, N. Afraz, H. Akse, K. Boodhoo, R. Bos, J. Cantin, Y. W. Chiang, J. M. Commenge, J. L. Dubois, F. Galli, J. Harmsen, S. Kalra, F. Keil, R. Morales-Mendez, F. J. Navarro-Brull, T. Noel, G. Patience, D. Reay, R. Santos, A. Smith-Schoettker, A. Stankiewicz, H. van den Berg, T. V. Gerven, J. van Gestel, M. van der Stelt, M. van de Ven, R. Weber, *Educ. for Chem. Eng.* **2020**, *32*, 1 .

[125] D. F. Rivas, K. Boodhoo, R. Morales-Mendez, Y. W. Chiang, F. Navarro, T. V. Gerven, J. van Gestel, R. Weber, S. Kalra, D. C. Boffito, D. Reay, J. A. F. Albanese, F. Keil, T. Noel, J. Glassey, A. Stankiewicz, H. van den Berg, F. Galli, K. Ogden, R. Santos, A. Smith-Schoettker, J. Cantin, H. Akse, G. Patience, J. M. Commenge, J. Harmsen, J. L. Dubois, N. Afraz, R. Bos, *Educ. for Chem. Eng.* **2020**, *32*, 15 .

[126] CHEMSAFE **2020**.
URL https://dechema.de/en/chemsafe.html

[127] A. Kummer, T. Varga, *J. Loss Prev. Process. Ind.* **2019**, *58*, 22 .

[128] J. E. Coon, M. Kush, M. C. Rowland, J. R. Cunningham, Presented at the AIChE Spring Meeting, Citeseer **1998**, pp. 1 – 11.

For Peer Review